

任务型对话方法、存储介质及计算机设备

技术领域

本申请涉及互联网技术领域，尤其是涉及到一种任务型对话方法、存储介质及计算机设备。

背景技术

任务型对话是一种旨在协助用户高效完成特定目标或任务的交互式对话方式。在此类对话中，人工智能系统致力于深入理解用户的意图，并据此提供针对性的信息、建议或执行相关操作，以此满足用户的多样化需求。目前，任务型对话广泛应用于企业自动化助手、客户服务支持、智能家居控制等多种领域，可以极大地提升工作效率、优化用户体验，并减少人工干预的需求。

当前，任务型对话主要依赖于意图识别、槽位填充和知识图谱等自然语言处理技术来实现。然而，上述方法在处理用户意图时，往往依赖于文本分类来理解对话中的核心需求，或者依赖知识图谱来寻找解决方案。这些方法要求系统能够预先枚举出用户可能遇到的所有问题及其对应的解决方案，然而，用户的个性化需求是极其多样且难以穷举的。因此，此类方法在样本数据的构建方面难度非常高。此外，这种基于枚举的方法往往也存在较大的滞后性，且方案匹配的准确度受限于预设的枚举范围，也难以满足用户不断变化的实际需求。

基于此，在应对一些较为复杂或专业化的任务时，用户往往需要转向人工服务寻求帮助，这不仅降低了问题解决的时效性，也增加了人力成本的投入。

发明内容

有鉴于此，本申请实施例提供了一种任务型对话方法、存储介质及计算机设备，主要目的在于解决现有的任务型对话在解决复杂和专业化任务时，成功率低、准确性差、时效性差的技术问题。

根据本申请的第一个方面，提供了一种任务型对话方法，该方法包括：

客户端响应于接收到用户发起的对话信息，将所述对话信息发送至服务器；

所述服务器响应于接收到所述对话信息，通过大模型确定所述用户的意图信息，并根据所述意图信息生成工具执行计划，执行所述工具执行计划，得到计划执行结果，对所述计划执行结果进行检查，并基于检查通过的计划执行结果生成回复内容，将所述回复内容发送至所述客户端；

所述客户端接收并显示所述回复内容。

依据本申请第二个方面，提供了一种存储介质，其上存储有计算机程序，所述程序被处理器执行时实现上述任务型对话方法。

依据本申请第三个方面，提供了一种计算机设备，包括存储介质、处理器及存储在存储介质上并可在处理器上运行的计算机程序，所述处理器执行所述程序时实现上述任务型对话方法。

借由上述技术方案，本申请实施例提供了一种任务型对话方法、存储介质及计算机设备，在接收到用户发起的对话信息时，首先通过大模型确定用户的意图信息，然后根据意图信息生成工具执行计划，并执行工具计划，得到计划执行结果，进而对计划执行结果进行检查，并基于计划执行结果生成回复内容，最后将回复内容发送至客户端进行展示。上述方法通过大模型进行意图识别、工具调用、工具执行、文本检查和文本总结，可以充分利用大模型的文本理解能力和文本生成能力，提高任务型对话进行意图识别和文本处理的时效性和准确度。相比于传统的案例匹配式的任务型对话方法，上述方法可以降低任务穷举的工作量，提升任务解决的成功率，从而使任务型对话能够更加灵活地应用于各个专业领域中，并用于解决更为复杂、专业化和个性化的任务。此外，上述方法通过将用户的对话信息拆解成为更简单、清晰的工具执行计划，可以将内容回复的范围限定在工具执行结果的范围内，从而可以避免大模型拆解的

任务过于发散而偏离主题, 以此提升任务执行的准确性, 同时, 也可以降低大模型进行任务分解和任务执行时的难度, 从而提高任务型对话的时效性。

附图说明

- 图 1 示出了本申请实施例提供的一种任务型对话方法的流程示意图;
图 2 示出了本申请实施例提供的另一种任务型对话方法的流程示意图;
图 3 示出了本申请实施例提供的一种任务型对话方法的系统架构示意图;
图 4 示出了本申请实施例提供的一种案例生成链路的系统架构示意图;
图 5 示出了本申请实施例提供的一种信息检索模型的系统架构示意图;
图 6 示出了本申请实施例提供一种精度和召回率随阈值变化的趋势图;
图 7 示出了本申请实施例提供的一种任务型对话装置的结构示意图;
图 8 示出了本申请实施例提供的另一种任务型对话装置的结构示意图。

具体实施方式

下文中将参考附图并结合实施例来详细说明本申请。需要说明的是, 在不冲突的情况下, 本申请中的实施例及实施例中的特征可以相互组合。

针对现有 Agent 架构在任务型对话实现中的局限性, 特别是在解决复杂或专业领域任务方面的不足, 本申请提出了一种基于 E-Agent 架构的任务型对话方法。该方法可以在速度、准确性、稳定性和模型易训练性之间找到较好的平衡, 从而可以更有效地满足任务型对话的实际需求。其中, E-Agent 架构包括意图识别模块、计划制定模块 (Planner)、调用执行模块 (Caller)、检查校验模块 (Checker) 和内容总结模块 (Summary), 各个模块之间协同工作, 可以快速、稳定的实现任务型对话, 并满足用户的个性化需求。对于 E-Agent 架构的具体实现方式, 将在以下任务型对话方法的各个实施例中进行具体阐述。

在一个实施例中, 如图 1 所示, 提供了一种任务型对话方法, 以该方法应用于客户端和服务器等计算机设备为例进行说明, 上述方法可以包括以下步骤:

步骤 101, 客户端响应于接收到用户发起的对话信息, 将对话信息发送至服务器。

具体的, 用户可以通过客户端 (如智能终端上安装的应用程序或者网页等) 上展示的信息输入框输入任意的对话信息并发送, 客户端在接收用户发起的对话信息后, 可以将对话信息封装为标准的格式后, 通过网络协议发送至服务器。在本实施例中, 信息输入框为能够接收文本信息的任意组件, 例如, 信息输入框可以为聊天输入框、搜索输入框等等, 对话信息可以为用户输入的任意文本信息, 同时, 也可以包括语音信息、视频信息、图片信息等其他形式的信息。

步骤 102, 服务器响应于接收到对话信息, 通过大模型确定用户的意图信息, 并根据意图信息生成工具执行计划, 执行工具执行计划, 得到计划执行结果, 对计划执行结果进行检查, 并基于检查通过的计划执行结果生成回复内容, 将回复内容发送至客户端。

其中, 大模型指的是大语言模型 (Large Language Model, LLM)、多模态模型 (Large Multimodal Models, LMMs) 等能够处理复杂任务和大数据量的高级人工智能模型。其中, 大模型可以是通用领域的大模型, 也可以是经过预设领域的样本数据进行参数优化后的垂直领域的大模型 (下述垂域模型)。在本实施例中, E-Agent 架构中的各模块可以基于大模型实现。在通过大模型实现各模块功能之前, 服务器可以对大模型进行继续预训练 (Continue pre-train)、监督式微调 (Supervised Fine-Tuning, SFT)、模型量化等多种形式的预处理, 以提高大模型任务执行的时效性、稳定性和准确性。可以理解的是, 本实施例对于 E-Agent 架构中各个模块所采用的大模型的具体类型不做具体限定。

具体的, 服务器在接收到用户发起的对话信息之后, 可以将获取到的对话信息作为参数, 输入到预训练的大模型中, 以使大模型能够基于对话信息, 准确的定位出用户的意图信息。在本实施例中, 在进行意图识别时, 大模型还可以获取当前对话的上下文信息、应用场景信息以及经过用户授权获取的用户信息等多种类型的信息, 以使大模型能够

更加准确的识别出用户的意图信息。在本实施例中，对于输入到大模型中进行意图识别的参数不做具体限定，其中，意图识别的过程可以通过 E-Agent 架构中的意图识别模块来实现。

进一步的，服务器在确定了用户的意图信息之后，可以将意图信息输入到另一个大模型中，以使大模型可以通过意图信息选取至少一个工具，以生成工具执行计划。在本实施例中，工具执行计划可以包含与意图信息相关的至少一个工具（即程序功能接口），其中，工具的选取范围可以预先划定或者通过前置的向量化粗召回环节得到。通过调用工具，即可实现工具对应的功能。例如，当意图信息为“查询商品销量”时，大模型选取出的工具可以为菜品数据查询工具“dish_data_query”，其中，菜品数据查询工具“dish_data_query”即为基于意图信息“查询商品销量”生成的工具执行计划。可以理解的是，在意图信息较为简单和明确时，工具执行计划也较为简单和明确；在意图信息较为复杂时，所需的工具也较多，且不排除有遗漏工具的可能性。因此，后续会对工具执行计划的计划执行结果进行检查，以确保工具无遗漏。在本实施例中，对于大模型选取的工具的范围和工具的具体功能均不做具体限定，其中，工具执行计划生成的过程可以通过 E-Agent 架构中的计划制定模块（Planner）来实现。

进一步的，在得到工具执行计划之后，大模型可以执行工具执行计划，得到计划执行结果。在本实施例中，工具执行计划的执行过程即为调用工具执行计划中各个工具的过程，在此过程中，大模型可以获取工具参数，并通过工具参数调用各工具，得到各工具的调用结果，后续，通过对各个工具的调用结果进行整合，即可得到计划执行结果。在本实施例中，工具所需的工具参数一般为用户 ID，商品 ID，订单 ID，时间范围等较为常规的信息，因此，工具参数通常可以在用户的对话信息、上下文信息和用户信息等信息中提取得到。在本实施例中，对于大模型提取工具参数的方式不做具体限定，其中，工具执行计划的执行过程可以通过 E-Agent 架构中的调用执行模块（Caller）来实现。

进一步的，在得到计划执行结果之后，大模型可以对计划执行结果的完备性进行检查，并通过计划执行结果的完备性，确定计划执行结果是否检查通过。若计划执行结果检查通过，则可以将计划执行结果输入到下一环节中，生成回复内容；若计划执行结果未检查通过，则需要通过计划制定模块重新选取工具，并更新工具执行计划，进而执行更新后的工具执行计划，得到更新后的计划执行结果，然后再判断更新后的计划执行结果是否完备。此过程可以不断循环，直至修改后的计划执行结果达到完备性要求和/或循环次数达到一定阈值后，再输出计划执行结果，最后通过更新后的计划执行结果生成回复内容。在本实施例中，大模型可以通过对话信息的内容对计划执行结果的完备性进行判断，在判断出与对话信息相关的工具已全部调用和/或计划执行结果能够解决对话信息中提出的问题这两个条件中的至少一个条件时，即可判定计划执行结果完备，此时，可以对计划执行结果进行输出，并用于生成回复内容。其中，计划执行结果的检查过程可以通过 E-Agent 架构中的检查校验模块（Checker）来实现。

进一步的，在计划执行结果检查通过后，大模型可以基于检查通过的计划执行结果生成回复内容，其中，回复内容可以结合用户输入的对话信息生成，即在生成回复内容时，可以对计划执行结果中与对话信息相关的内容进行保留，并将与对话信息无关的内容进行剔除，以此得到与对话信息较为相关的回复内容。在本实施例中，大模型还可以对回复内容进行其他形式的处理，如增加安抚性信息、增加链接信息等等，本实施例在此不做具体限定。其中，回复内容的生成过程可以通过 E-Agent 架构中的内容总结模块（Summary）来实现。

在本实施例中，通过基于用户的对话信息生成用户的意图信息，可以明确用户的任务需求，通过基于意图信息生成工具执行计划，可以将复杂或者个性化的任务需求拆解为简单的工具调用列表，从而可以减少文本信息处理的复杂度，相比于传统的任务拆解方式，制定工具执行计划可以避免任务拆解的过于分散或者冗长，从而可以提高信息处理的集中度，降低复杂任务出错的概率。进一步的，通过执行工具计划，并对计划执行结果进行检查，可以提高文本信息的检查能力和反馈能力，降低信息遗漏的可能性，最后，通过基于检查通过的计划执行结果生成回复内容，可以提升回复内容的针对性和专业性。

步骤 103，客户端接收并显示回复内容。

具体的，在生成回复内容之后，服务器可以将回复内容发送至客户端，客户端响应于接收到回复内容，可以对回复内容进行展示。例如，客户端可以通过即时通讯页面、搜索结果页面等多种页面对回复内容进行展示。在本实施例

中，服务器在输出回复内容时，还可以输出其他信息，例如，服务器还可以输出用户的对话信息、意图信息、工具名称、工具调用结果、大模型的思考信息等等。在此场景下，服务器可以仅将大模型输出的部分信息发送至客户端进行展示，例如，服务器可以仅将回复内容发送至客户端，然后将其他信息记录在日志中，以便于日后进行分析处理。

进一步的，作为上述实施例具体实施方式的细化和扩展，为了完整说明本实施例的具体实施过程，提供了另一种任务型对话方法，该方法可以应用于如图3和图4所示的E-Agent架构中，如图2所示，该方法可以包括以下步骤：

步骤201，客户端响应于接收到用户发起的对话信息，将对话信息发送至服务器。

步骤202，服务器响应于接收到对话信息，获取上下文信息，并基于对话信息和上下文信息，通过大模型确定用户的意图信息。

在本实施例中，在进行意图识别时，大模型还可以获取应用场景信息以及经过用户授权获取的用户信息等多种类型的信息，以使大模型能够更加准确的识别出用户的意图信息。在本实施例中，对于输入到大模型中进行意图识别的参数不做具体限定，其中，意图识别的过程可以通过E-Agent架构中的意图识别模块来实现。

在一个可选的实施方式中，步骤202可以通过以下方式实现：服务器响应于接收到对话信息，获取对话信息的上下文信息和预设的应用场景信息，并将应用场景信息作为约束条件，基于对话信息和上下文信息确定用户的意图信息。

具体的，服务器在接收到用户的对话信息后，可以先获取对话信息的上下文信息和预设的应用场景信息，其中，上下文信息可以包括用户此前的历史对话信息以及会话状态等信息，应用场景信息可以根据对话发生的场景、用户所使用的服务类型或特定领域的信息处理逻辑设定。例如，在商家助手这一场景中，应用场景信息即为商家服务的应用场景，在该场景下，所有任务均围绕着为商家服务这一核心主题展开。进一步的，服务器可以将获取到的应用场景信息作为约束条件，结合对话信息和上下文信息，输入到预先训练好的大模型中，以通过大模型准确理解上述信息中的自然语言，并推断出用户的核心意图。

在上述实施方式中，大模型可以综合考虑对话信息、上下文信息以及应用场景的约束条件，得出较为符合当前应用场景的用户意图信息。例如，在商家服务场景下，商家用户通常存在表达内容口语化、对话内容冗长、对话内容拆分多句发送等表达习惯，此时，若将原始的多轮对话直接用于工具匹配和案例搜索，则在工具和案例的召回精度上均存在较大挑战。基于此，本实施方式通过设定一个多轮改写模块，可以利用大模型强大的自然语言理解能力，关联上下文转写出当前轮次的核心意图，最终输出表达清晰明确的目标（Goal）。

在一个可选的实施方式中，服务器可以通过以下至少一种方式确定用户的意图信息：

202-1，服务器在应用场景信息约束的话题范围内，对对话信息和上下文信息进行改写，得到用户的至少一条意图信息。

具体的，服务器在进行意图识别时，可以在应用场景信息约束的话题范围内，对用户的对话信息和上下文信息进行改写。其中，改写方式可以包括但不限于同义词替换、语义扩展、词句重组等等，以此确保改写后的内容能够符合应用场景的约束条件。在本实施例中，针对识别出的意图信息主题的不同，大模型可以以列表的形式输出至少一条意图信息，以便于大模型能够准确的选取所有可以利用的工具，从而制定出较为完备的工具执行计划。

202-2，服务器在判断出对话信息和上下文信息未在应用场景信息约束的话题范围内时，将对话信息作为用户的意图信息。

具体的，服务器在判断出对话信息和上下文信息未在应用场景信息约束的话题范围内时，可以不再对对话信息进行改写操作，而是直接将对话信息作为用户的意图信息。此时，服务器可以根据用户实际输入的对话信息进行后续处理，如提示用户重新输入或引导用户输出更有效的任务需求等等。

202-3，服务器基于对话信息和上下文信息，通过预设的信息检索模型，在意图信息经验库中搜索出至少一个案例，并将案例设置在提示模版中，以通过提示模版确定用户的意图信息。

具体的，服务器还可以通过预设的信息检索模型搜索出相似的案例，并在相似案例的辅助下进行意图识别。其中，信息检索模型可以包括稀疏检索模型、稠密向量检索模型和多向量检索模型中的至少一种模型。服务器通过将对话信

息和上下文信息作为查询条件输入到上述信息检索模型中，可以在意图信息经验库中搜索出至少一个与查询条件相关的案例。然后，服务器可以将这些案例设置在大模型的提示模版中，并根据提示模版的内容确定用户的意图信息。在本实施例中，意图信息经验库中可以预先存储有多种场景下的意图识别案例，其中，意图识别案例的输入为上下文信息和对话信息连接而成的多轮对话信息，输出为与多轮对话信息对应的单轮意图信息。

在一个可选的实施方式中，在确定用户的意图信息时，还可以通过以下至少一种方式对用户的对话信息和上下文信息进行改写：

202-1-1，服务器对对话信息和上下文信息中与应用场景信息相关的信息进行保留，并根据保留的内容的主题生成至少一条意图信息。

具体的，服务器可以识别出对话信息和上下文信息中与应用场景信息紧密相关的信息。这些信息可以是特定领域的术语、概念或流程等等。通过对这些信息进行保留，并根据保留下的内容的主题进行改写，可以确保生成的意图信息既简洁又明确。例如，对话信息和上下文信息如下：商家：“我昨天是不是有好评？”；助手：“经查询您昨天是有一个好评：“菜品非常合我口味，赞。”；商家：“我能置顶吗？我想让更多的消费者看到这条消息。”。经过改写后，可以保留以下两条意图信息：“怎么置顶昨天的好评？”，“如何让更多的人看到昨天的好评？”。通过上述改写，可以有效的明确商家用户的核心意图。

上述实施方式通过在复杂的对话信息和上下文中抽取与应用场景信息相关的核心意图，可以使生成的意图信息更为准确，易于后续处理。同时，上述方式也有助于提高系统对于特定领域的理解能力，从而更好地满足用户的需求。

202-1-2，服务器将对话信息和上下文信息中的时间信息和数字信息保留在用户的意图信息中。

具体的，服务器可以抽取对话信息和上下文信息中的时间信息和数字信息。其中，时间信息和数字信息可以用于明确用户的核心意图，且可以作为工具参数便于后续提取。服务器通过将这些信息保留在改写后的意图信息中，可以确保意图的完整性和准确性。例如，当商家用户询问“帮我分析下最近一周的好评”时，服务器可以将“最近一周”这一时间信息保留在意图信息中。

202-1-3，服务器将对话信息和上下文信息中的指代信息替换为名称信息保留在用户的意图信息中。

具体的，服务器可以针对对话信息和上下文信息中的指代信息进行处理，指代信息指的是用户在对话中使用的代词、名词短语等，这些指代信息可以结合上下文信息来明确其指代的对象。基于此，服务器可以通过分析上下文信息，将指代信息替换为明确的名称信息，并将其保留在改写后的意图信息中。例如，如果商家用户询问“怎么联系上他？”，服务器可以识别出“他”指的是某个订单的配送人员，然后将“他”替换为该订单的配送人员后保留在意图信息中。

上述实施方式通过将指代信息替换为明确的名称信息，可以使改写后的意图信息能够更清晰地表达出用户的实际需求。上述改写方式可以有助于消除指代信息带来的歧义性和不确定性，从而提高系统响应的准确性和可靠性。

202-1-4，服务器在识别出上下文信息中包含与对话信息不相同的差异性话题时，将与差异性话题相关的内容在意图信息中删除。

具体的，服务器可以识别出上下文信息中与当前轮次的对话信息不同的差异性话题。其中，差异性话题指的是用户在同一个会话中更换的主题或话题，与差异性话题相关的信息可能会干扰系统对核心意图的理解。基于此，可以将与这些差异性话题相关的内容在改写后的意图信息中删除。例如，如果商家用户先询问了关于某产品的问题，然后又询问了关于某订单的问题，则服务器会识别出这两个话题的差异，然后在生成意图信息时删除与上述产品相关的内容。

上述实施方式通过删除与差异性话题相关的内容，可以使改写后的意图信息能够更专注于当前的核心意图，这种改写方式有助于减少系统对不必要信息的处理量，提高响应速度和准确性。同时，由于删除了与差异性话题相关的内容，可以使得改写后的意图信息更易于后续的处理和分析。

在一个可选的实施方式中，意图识别的过程可以通过在意图识别模块的大模型中设置提示模板来实现，例如，大模型的提示模板中可以设置如下内容：

结合历史对话和应用场景，以商家身份重新表述商家当前问题，使表述更清晰易理解；

- 1、思考历史对话和应用场景是否有关，如果无关，商家当前问题不做改写；
- 2、问候、批评、赞叹等闲聊场景的，商家当前问题不做改写；
- 3、重新表述后，不能缺省问题相关的时间和订单号条件；
- 4、重新表述的结果中禁止自己臆想的条件。

nResult: 以商家身份表述商家问题，多意图需分解成多个独立问题的列表。

案例 1：商家：我昨天是不是有好评？助手：经查询您昨天是有一个好评：“菜品非常合我口味，赞”。商家：我能置顶吗？我想让更多的消费者看到这条消息。Thought: 商家说的置顶对象是上文中的好评记录，且问题有时间条件：昨天。Result: “怎么置顶昨天的好评？”，“如何让更多的人看到昨天的好评？”。

在一个可选的实施方式中，意图信息可以通过预训练的大模型生成，其中，上述大模型的预训练方法可以通过以下方式实现：服务器基于预设领域的样本数据集，对通用领域的大模型进行继续预训练，其中，样本数据集中设置了随机噪声，且不包含样本用户信息，然后，基于预设领域的意图识别案例，对继续预训练后的大模型进行参数微调，最后，基于微调后的大模型生成预设领域中预设问题的意图识别信息，并基于意图识别信息的正负反馈信息，通过强化学习算法迭代更新大模型的参数，得到预训练后的大模型。

具体的，服务器在进行模型训练时，可以加载一个通用领域的大模型作为基座模型，然后使用预设领域的样本数据集对该基座模型进行继续预训练（Continue pretrain）。该数据集中可以包含特定领域内的任务对话数据，同时，样本数据中还可以加入随机噪声，以增强模型的鲁棒性，同时确保数据集中不包含任何样本用户信息，以保护用户的隐私。完成继续预训练后，服务器可以基于预设领域的意图识别案例对模型进行参数微调（SFT）。其中，意图识别案例可以包括预设领域的多轮对话信息及其对应的意图识别结果，这些案例可用于指导模型识别特定领域内的用户意图。在微调的过程中，大模型可以根据案例的输入和预期输出来调整其内部参数，以优化大模型在特定领域内的性能。随后，服务器可以利用微调后的模型生成预设领域中预设问题的意图识别信息。这些意图识别信息可以被用于与用户进行交互，并根据用户的反馈进行迭代优化。具体地，服务器可以收集用户对意图识别信息的正负反馈信息，并通过强化学习算法（RLHF）迭代更新模型的参数。其中，强化学习算法可以根据用户的反馈来调整模型的行为策略，以使得模型能够更准确地识别用户的意图。最后，可以通过深度伪强化学习（DPO）等技术进一步提升模型的性能，从而得到预训练后的大模型。经过预训练后的模型不仅具备在特定领域内准确识别用户意图的能力，而且还能够在与用户交互的过程中持续学习和优化。

步骤 203，服务器基于意图信息选取至少一个工具，生成工具执行计划。

具体的，服务器在确定了用户的意图信息之后，可以将意图信息输入到大模型中，以使大模型可以通过意图信息选取至少一个工具，从而得到工具执行计划。在本实施例中，工具执行计划可以包含与意图信息相关的至少一个工具（即程序功能接口），其中，工具的选取范围可以预先划定或者通过前置的向量化粗召环节得到。通过调用工具，即可实现工具对应的功能。例如，当意图信息为“查询商品销量”时，大模型选取出的工具可以为菜品数据查询工具“dish_data_query”，其中，菜品数据查询工具“dish_data_query”即为基于意图信息“查询商品销量”生成的工具执行计划。可以理解的是，在意图信息较为简单和明确时，工具执行计划也较为简单和明确；在意图信息较为复杂时，所需的工具也较多，且不排除有遗漏工具的可能性。因此，后续会对工具执行计划的计划执行结果进行检查，以确保工具无遗漏。在本实施例中，对于大模型选取的工具的范围和工具的具体功能均不做具体限定，其中，工具执行计划生成的过程可以通过 E-Agent 架构中的计划制定模块（Planner）来实现。

在一个可选的实施方式中，步骤 203 可以通过以下方式实现：服务器获取上下文信息，并根据上下文信息，在工具库中查找与意图信息相关的多个候选工具，生成候选工具列表，然后基于意图信息，在候选工具列表中选取至少一个工具，并对选取的工具进行排列组合，得到工具执行计划。

具体的，服务器可以根据上下文信息，在预先构建的工具库中以查找与上下文信息相关的多个候选工具，这一过程可以前置在 Planner 模块选取工具之前完成，并作为粗召环节实施。该过程可以通过匹配上下文信息与工具库中工

具的描述、功能或标签等信息来完成，从而得到一个候选工具列表。进一步的，服务器可以利用预设的算法或规则，基于意图信息的具体内容，在候选工具列表中选择出至少一个工具。这些工具可以按照完成任务过程中的逻辑关系和依赖关系进行排列组合，以形成一个或多个工具的工具执行计划。需要说明的是，上述工具执行计划的生成方式不仅适用于商家服务场景，同时也具有较强的通用性和可扩展性，可以广泛应用于其他需要工具辅助完成任务的场景中。

上述实施方相比于 XAgent 架构中直接拆解任务的方式，可以避免任务拆解过于发散和任务列表冗长的问题。同时，相比于 ReWOO 架构一次性生成完整计划的方式，可以通过精确匹配意图信息和候选工具的方式，提高工具选择的准确度，从而提升了整个计划执行的可行性和执行效率。此外，通过在前置环节中加入工具的向量化粗召环节，能够更快速地定位到与意图信息最相关的工具集，从而能够提升工具选取的效率和准确性。

在一个可选的实施方式中，服务器可以通过以下至少一种方式选取工具：

203-1，服务器基于意图信息，在候选工具列表中进行关键词匹配，筛选出至少一个工具。

具体的，服务器可以基于意图信息以及工具的描述、标签等信息，在预先构建的候选工具列表中通过关键词匹配算法，快速的筛选出与意图信息相关的至少一个工具。其中，用于匹配的关键词可以来源于用户的意图信息，也可以是经过处理（如分词、去停用词等）后提取出的其他关键元素。

上述实施方式通过采用关键词匹配算法选取工具，可以快速定位到与用户意图直接相关的工具，从而提高工具选取的效率和准确性。同时，上述方式不需要复杂的自然语言处理过程，可以降低系统的复杂性以及对计算资源的要求。

203-2，服务器在判断出当前轮次的意图信息与上一轮次的意图信息含义相同时，将与上一轮次的工具不同的工具作为当前轮次选取的工具。

具体的，服务器可以记录用户的历史意图信息。当判断出当前轮次的意图信息与上一轮次（或者上几个轮次）的意图信息含义相同时，服务器可以避免重复选择相同的工具，而是在上一轮次已使用的工具中增加新的工具作为当前轮次的选择，或者替换掉上一轮次已使用的工具作为当前轮次的选择。这种工具选取策略有助于在连续的任务中提供多样化的工具支持，增强用户的体验感。

203-3，服务器将与选取的工具互斥的工具作为当前轮次禁用的工具。

具体的，服务器可以维护一个工具互斥规则库，该规则库可以包含工具之间的互斥关系。当服务器在选取工具时，可以检查选取的工具是否与已选工具存在互斥关系。如果存在互斥关系，服务器会将该工具作为当前轮次禁用的工具，以确保不会选取到与已选工具冲突的工具。

203-4，服务器在基于意图信息未选取工具时，将知识检索工具作为当前轮次选取的工具。

具体的，在基于意图信息未能选取出合适的工具时，服务器可以采取一种备选策略，即将知识检索工具作为当前轮次选取的工具。其中，知识检索工具在特定领域中具有强大的信息搜索和整合能力，可以帮助服务器在更专业的知识库中查找与意图信息相关的信息或资源，从而为后续的任务执行提供支持。

203-5，服务器基于意图信息，通过预设的信息检索模型，在执行计划经验库中搜索出至少一个案例，并基于案例在候选工具列表中选择出至少一个工具。

其中，信息检索模型可以包括稀疏检索模型、稠密向量检索模型和多向量检索模型中的至少一种模型。

具体的，服务器可以基于用户的意图信息，利用预设的信息检索模型（如稀疏检索模型、稠密向量检索模型或多向量检索模型），在执行计划经验库中搜索出与意图信息相似的历史案例。这些历史案例包含了成功完成类似任务的经验和工具组合信息。通过分析这些案例，服务器可以在候选工具列表中选择出与案例中使用的工具相似的至少一个工具。

在一个可选的实施方式中，服务器可以通过以下至少一种方式基于搜索出的案例选取工具：

203-5-1，服务器在案例与意图信息的匹配度大于预设阈值时，将案例中的工具作为选取的工具。

203-5-2，服务器在案例与意图信息的匹配度小于或等于预设阈值时，将案例设置在提示模版中，以通过提示模版在候选工具列表中选择出至少一个工具。

具体的，服务器在利用预设的信息检索模型在预先构建的执行计划经验库中搜索出与意图信息相关的案例之后，可以计算每个案例与意图信息之间的匹配度，并将计算出的匹配度与预设的阈值进行比较。如果搜索出的案例与意图信息的匹配度大于预设阈值，则可以采用案例中所使用的工具作为选取的工具。此时，高匹配度的案例往往意味着其使用的工具与当前意图信息高度相关，能够很好地满足用户的需求。相反，如果选取出的案例与意图信息的匹配度小于或等于预设阈值，则可以将搜索到的案例设置在大模型的提示模版中，以使大模型根据提示模版中的案例的描述和相关信息，在候选工具列表中选取至少一个工具。这种工具选取方式允许用户基于部分匹配的案例进行进一步的判断和选择，从而提高工具选取的灵活性和准确性。

上述实施方式通过基于案例与意图信息之间的匹配度选取工具，可以有效的提升工具选取的准确性和选取效率。

在一个可选的实施方式中，服务器可以通过以下方式对选取的工具进行排列组合，得到工具执行计划：服务器基于选取的各工具的工具参数之间的依赖关系，对工具进行顺序排列和/或并行排列，然后通过预设的运算符，对顺序排列和/或并行排列的工具进行运算连接，得到工具执行计划。

具体的，服务器可以基于用户的意图信息在工具库中选取至少一个工具。例如，在商家服务领域中，服务器可以选取的工具包括但不限于知识检索工具、经营诊断、查询商家 XX 基础信息、查询商家 XX 经营数据，以及设置 XX 功能的写操作接口等多种工具，总计可达上百种以上。在选取适合的工具之后，服务器可以分析每个工具的工具参数之间的依赖关系。这些依赖关系可能基于数据流的连续性、操作的前后顺序要求或数据处理逻辑的要求等等。服务器可以根据这些依赖关系，对选取出的工具进行顺序排列和/或并行排列。其中，顺序排列的工具需要按照特定的顺序依次执行，以保证数据的准确性和操作的连贯性；而并行排列的工具可以并行执行，以提高处理效率。在完成工具排列后，服务器可以利用预设的运算符，如括号、逗号等，对顺序排列和/或并行排列的工具进行运算连接。其中，运算符不仅可以定义工具的执行顺序，同时也确保整个工具执行计划的逻辑正确性。通过运算符的连接，服务器最终可以生成如“[(T1, T2), (T3)]”的工具执行计划，其中，T1 与 T2 代表需要顺序执行的工具，而 T3 则代表可以与 T1 和 T2 并行执行的工具。

在一个可选的实施方式中，工具执行计划可以通过在计划制定模块的大模型中设置提示模板来实现，例如，大模型的提示模板中可以设置如下内容：

你是一个外卖平台的智能机器人，你叫商家小助手，你将拥有商家端 APP 最高权限与商家在 APP 沟通。你的核心目标是理解和解决商家的问题，提高他们的客户满意度，目前职责是根据商家问题选择需要调用的多个工具。

目标：必须正确分析出商家的意图，并根据这个意图在提供的工具列表中选择有用的工具。以下 API 工具可以使用：

unclear_intention：代表意图不清晰或者无意义语气词；unrelated_biz：代表商家意图与外卖场景无关；Question：用户问题；Thought：请仔细阅读用户和你的聊天记录，分析用户问题，以便于选择出最适合商家问题的工具列表；Action：最适合解决问题的工具列表，不可为空，多个工具使用英文逗号隔开，选择的工具必须在{tool_names}之内；Observation：调用工具后给出的反馈。

注意事项：

*必须优先使用工具，如果其他工具无法匹配和使用，则使用“knowledge_retrieval_tool”；

*注意当用户重复咨询一个问题，可能是上一次调用工具错误，请酌情更换工具；

*工具是互斥，比如开启和关闭。请勿同时调用。比如 open 和 close；

*如果商家的问题是无意义语气词或者输入不完整你必须使用“unclear_intention”；

*禁止猜测商家意图，禁止选择没有提供的工具，禁止不选工具，禁止直接回答，禁止等待商家的新问题；

案例 1：Question：如何绑定 XX（注：XX 用于代指某即时通讯应用软件）。Thought：XX 和外卖场景没有直接关联，但是作为线上外卖平台，可能和 XX 通信工具有关联，尝试使用“knowledge_retrieval_tool”检索知识回答。
Action：knowledge_retrieval_tool。

在一个可选的实施方式中，工具执行计划可以通过预训练的大模型生成。其中，上述大模型的预训练方法可以通过以下方式实现：服务器基于预设领域的样本数据集，对通用领域的大模型进行继续预训练，然后基于预设领域的工具调用案例，对继续预训练后的大模型进行参数微调，最后将微调后的大模型的参数权重从浮点数转换为预设位数的整数，得到预训练后的大模型。

具体的，服务器在进行模型训练时，可以加载一个通用领域的大模型作为基座模型，然后使用预设领域的样本数据集对该基座模型进行继续预训练（Continue pretrain）。其中，该数据集中可以包含特定领域内的工具调用数据，通过上述训练过程，可以使模型更好地适应特定领域的分布和语言模式。在继续预训练完成后，服务器可以进一步基于预设领域的工具调用案例，对继续预训练后的大模型进行参数微调（SFT）。这些工具调用案例可以包含用户问题与工具之间的映射关系，从而能够指导模型学习如何从用户问题中准确识别出需要调用的工具。通过模型微调，大模型能够更好地理解用户问题的意图，并精准地映射到对应的工具上。微调完成后，服务器可以将大模型的参数权重从浮点数转换为预设位数的整数（int4 量化），该步骤能够在一定程度上保持模型的性能，同时大幅降低模型的复杂度和运行时的计算资源消耗。最后，经过以上步骤处理得到的大模型，可以被用于生成工具执行计划。当用户提出需求时，该模型能够根据用户的输入问题，准确地生成对应的工具执行计划。

步骤 204，服务器在上下文信息中提取出工具参数，并基于工具参数对工具执行计划中的工具进行调用，得到计划执行结果。

具体的，在得到工具执行计划之后，大模型可以执行工具执行计划，得到计划执行结果。在本实施例中，工具执行计划的执行过程即为调用工具执行计划中各个工具的过程，在此过程中，大模型可以获取工具参数，并通过工具参数调用各工具，得到各工具的调用结果，后续，通过对各个工具的调用结果进行整合，即可得到计划执行结果。在本实施例中，工具所需的工具参数一般为用户 ID，商品 ID，订单 ID，时间范围等较为常规的信息，因此，工具参数通常可以在用户的对话信息、上下文信息和用户信息等信息中提取得到。在本实施例中，对于大模型提取工具参数的方式不做具体限定，其中，工具执行计划的执行过程可以通过 E-Agent 架构中的调用执行模块（Caller）来实现。

在一个可选的实施方式中，步骤 204 可以通过以下方式实现：服务器获取上下文信息，并在对话信息和上下文信息中提取出工具执行计划中的工具所需的工具参数，然后基于工具参数，按照工具执行计划中的工具的排列组合的顺序，对工具进行串行调用和/或并行调用，得到计划执行结果。

具体的，服务器可以在上下文信息中提取出工具执行计划中的各工具的工具参数，其中，工具参数例如可以包括时间参数、商品 ID、订单 ID 等各类数据。通过提取工具参数，可以确保工具执行计划中的各个工具能够正确、高效的执行。在工具参数确定之后，服务器可以根据工具执行计划中定义的工具排列组合顺序，对工具进行调用。在很多领域中，如商家服务领域中，大部分工具之间并不存在明确的依赖关系，此时，服务器可以优先采用并行调用的方式，以最大化减少工具整体执行的延时。同时，在工具之间存在明确的依赖关系的情况下，服务器也可以按照依赖顺序进行串行调用，以确保工具的正确执行。

上述实施方式通过并行调用工具执行计划中的工具，能够在较大程度上减少工具整体执行的延时，从而提高任务执行的响应速度和工作效率。此外，通过基于对话信息和上下文信息自动提取工具参数，能够确保准确的调用各个工具，以此降低操作错误的风险，提高系统的自动化和智能化水平。

在一个可选的实施方式中，服务器还可以通过以下至少一种方式获取工具参数：

204-1，服务器在识别出存在缺失的工具参数时，将缺失工具参数的工具的执行顺序后置，并在已执行的工具的执行结果中提取缺失的工具参数。

具体的，当服务器在执行工具执行计划，发现某个工具所依赖的入参在当前上下文中缺失时，可以触发容错机制，即服务器可以将该工具的执行顺序后置，这样可以让前置工具有机会在执行过程中产生或提供缺失的参数。在当前置工具执行完毕后，服务器可以再次检查是否获得了新的参数信息。如果获得了缺失的参数，则服务器可以继续执行原计划中的工具；否则，服务器可以考虑采用其他策略，如询问用户获取工具参数或根据已有信息进行部分执行等等。

上述实施方式通过结合参数自动提取策略和参数提取容错机制，可以显著的提高工具执行计划的鲁棒性和工具执行的成功率，并且，上述方法可以有效的减少不必要的用户交互，提高工具执行的效率和成功率。

204-2，服务器基于缺失的工具参数，发起询问信息，并在用户对询问信息的答复信息中提取缺失的工具参数。

具体的，当服务器识别出存在缺失的工具参数时，除了触发容错机制之外，也可以基于缺失的参数向用户发起询问信息，并通过用户界面展示给用户。用户可以通过回复该询问信息的方式来提供缺失的参数。服务器在接收到用户的答复后，可以从中提取出缺失的参数，并继续执行原计划中的工具。

上述实施方式通过在计划执行的过程中主动询问用户获取参数，可以确保计划能够持续稳定的执行。这种动态获取参数方式不仅可以提高工具执行的效率，而且可以提高计划执行的成功率，避免计划中途执行失败，避免资源浪费。

在一个可选的实施方式中，服务器可通过以下至少一种方式处理工具参数：

204-3，服务器在工具参数中提取出时间参数，并将时间参数中的相对时间修改为绝对时间。

具体的，在服务器提取出的时间参数中，可能包含用户以相对时间的表述，如“过去三天”或“上周”等。为了增强时间转换的准确率，服务器引入了一个日期专用工具。该工具能够将用户表述的相对时间转换成精确的绝对时间。例如，将“过去三天”转换为具体的日期范围“start: 20240510, end: 20240512”。完成时间参数的转换后，服务器可以将更新后的工具参数用于后续的工具调用。

204-4，服务器在工具参数中提取出订单参数，并将订单参数中的用户表述修改为预设领域的标准表述。

具体的，服务器可以在工具参数中提取出与订单相关的参数，从而可以将用户表述转换为预设领域的标准表述。例如，服务器可以将用户表述“XX号订单”转换为标准表述“订单流水号XX”。这样，无论用户如何表述订单信息，服务器都可以将其映射到统一的标准表述上，以确保工具的正确调用。

上述实施方式通过对订单参数进行标准化处理，可以实现用户表述与工具调用之间的有效匹配。这不仅可以提高工具调用的准确性，同时也可以降低因用户表述差异导致的误操作风险。

在一个可选的实施方式中，工具执行计划中的工具可以包括知识库搜索工具、信息查询工具、功能设置工具、信息解析工具中的至少一种工具。在本实施方式中，通过预先构建知识库以及知识库搜索工具，可以将回复内容限制在预设的领域中，从而降低回复内容出错的风险。此外，通过结合信息查询工具、功能设置工具、信息解析工具等多种类型的工具，可以实现信息查询、信息解析、功能设置等多种类型的功能，以此丰富任务型对话的功能性。可以理解的是，本实施例对于各工具实现的具体功能以及应用领域的范围均不做具体限制。

在一个可选的实施方式中，服务器基于工具参数对工具执行计划中的工具进行调用，得到计划执行结果的过程可以通过以下方式实现：服务器基于工具参数，对工具执行计划中顺序排列的工具进行串行调用，并对并行排列的工具进行并行调用，得到计划执行结果。

具体的，服务器可以根据工具的执行顺序和依赖关系，对顺序排列的工具进行串行调用，并对并行排列的工具进行并行调用。在调用的过程过程中，服务器还可以实时监控工具的执行状态，并在必要时进行干预和调整。最终，服务器可以收集所有工具的执行结果，并生成一个综合的计划执行结果。

步骤 205，服务器基于对话信息和上下文信息，对计划执行结果进行检查。

具体的，在得到计划执行结果之后，大模型可以对计划执行结果的完备性进行检查，并通过计划执行结果的完备性，确定计划执行结果是否检查通过。若计划执行结果检查通过，则可以将计划执行结果输入到下一环节中，生成回复内容；若计划执行结果未检查通过，则需要通过计划制定模块重新选取工具，并更新工具执行计划，进而执行更新后的工具执行计划，得到更新后的计划执行结果，然后再判断更新后的计划执行结果是否完备。此过程可以不断循环，直至修改后的计划执行结果达到完备性要求和/或循环次数达到一定阈值后，再输出计划执行结果，最后通过更新后的计划执行结果生成回复内容。在本实施例中，大模型可以通过对话信息的内容对计划执行结果的完备性进行判断，在判断出与对话信息相关的工具已全部调用和/或计划执行结果能够解决对话信息中提出的问题这两个条件中的至少

一个条件时，即可判定计划执行结果完备，此时，可以对计划执行结果进行输出，用于生成回复内容。其中，计划执行结果的检查过程可以通过 E-Agent 架构中的检查校验模块（Checker）来实现。

在一个可选的实施方式中，步骤 205 可以通过以下方式实现：

205-1，服务器获取上下文信息，并基于对话信息和上下文信息，对工具执行计划中的工具和计划执行结果进行检查。

205-2，当服务器判断出工具执行计划中不存在缺失的工具，和/或，计划执行结果完备时，确定计划执行结果检查通过。

具体的，服务器在对计划执行结果进行检查时，可以首先获取与对话信息相关的上下文信息，然后根据获取到的上下文信息和对话信息，对工具执行计划中的工具以及计划执行结果本身的完备性进行检查。其中，检查过程主要涉及两个方面：一是确认工具执行计划中是否包含了所有必要的工具，没有遗漏；二是评估计划执行的结果是否完备，能否满足生成回复内容的要求。上述两个条件只要满足其中至少一个条件，即可判定计划执行结果检查通过。

在本实施例中，回复内容被限定在工具执行结果的范围内，而工具执行结果是由工具调用得到的，工具的选取是有范围限制的，这就导致与对话信息和上下文信息相关的所有工具都调用后，生成的计划执行结果可能依然无法满足生成回复内容的要求。在此情况下，如果检查出工具执行计划中已经包含了所有必要的工具，则认为大模型已达到问题解决的边界，即便此时没有解决用户提出的问题，但依然可以认为计划执行结果检查通过，以此避免大模型随意扩展其他内容而导致超过预设领域的话题边界，使得回复内容出现错误和偏差。在另外一些情况下，如果检查出存在未调用的工具，但是计划执行结果本身能够满足生成回复内容的要求，也可以认为计划执行结果检查通过。通过这种方式，可以减少计算资源的消耗，降低系统延时，提高问题解决的时效性。基于此，只要上述两个条件满足其一，即可认为计划执行结果检查通过。

上述实施方式通过工具调用的完备性和工具调用结果的完备性这两个方面对工具执行计划进行检查，可以确保计划执行结果的完整性，从而避免因工具遗漏或执行结果不完整而导致的回复错误或信息缺失的问题。上述方式不仅提升了用户的使用体验，同时也增强了系统的可靠性和稳定性。此外，上述方式能够在问题出现的早期阶段进行识别和纠正，从而降低回复内容出错的概率。

在一个可选的实施方式中，步骤 205 还包括以下步骤：

205-3，服务器在判断出工具执行计划中存在缺失的工具，和/或，计划执行结果不完备时，可以循环执行以下过程，直至修改后的计划执行结果检查通过，和/或，计划执行结果的检查次数达到预设阈值：服务器基于工具执行计划和计划执行结果的检查结果生成反馈信息，并基于反馈信息和意图信息修改工具执行计划，然后执行修改后的工具执行计划，得到修改后的计划执行结果。

具体的，服务器在判断出工具执行计划中未包含所有必要的工具，或者，判断出计划执行的结果不完备时，可以判定计划执行结果检查不通过。此时，checker 可以生成一个包含思考和反馈信号的文本格式信息（即反馈信息），并将这个信息反馈给 planner 模块。planner 模块在接收到反馈信息后，可以基于这个信息以及用户的意图信息，重新选择或调整工具（一般以增量的方式选取工具），并生成修改后的工具执行计划。这个修改后的计划随后会交给 caller 模块执行，并生成新的计划执行结果。然后，这个新的计划执行结果会再次被送到 checker 模块进行检查。这个过程会反复进行，直到 checker 模块检查通过，即确认信息完备且足够生成回复内容，或者达到预设的检查次数阈值上限。

在本实施例中，如果检查出工具执行计划中未包含所有必要的工具，则认为大模型还未达到问题解决的边界，此时，无论计划执行结果是否能够解决用户提出的问题，都可以认定计划执行结果检查未通过，以此提升计划执行结果的全面性。在另外一些情况下，如果检查出计划执行结果无法满足生成回复内容的要求，也可以认定计划执行结果检查未通过，并通过工具再次搜索问题解决途径，以确保问题解决的有效性。基于此，只要上述两个条件满足其一，即可认为计划执行结果检查不通过。可以理解的是，对于计划执行结果是否通过的具体条件，可以根据产品的实际形态进行设计，本实施例在此不做具体限定。

上述实施方式通过根据 checker 的反馈信息和用户的意图，智能地调整工具选择和执行计划，可以实现信息的自动补全，从而使得系统在面对复杂多变的对话场景时，能够保持较高的准确率和响应速度。同时，通过设定检查次数的阈值上限，系统能够在一定程度上控制循环的次数，避免因特殊情况导致的无限循环，从而增强了系统的鲁棒性，同时，也提高了系统的响应时效性。

在一个可选的实施方式中，步骤 205-3 中基于反馈信息和意图信息修改工具执行计划的方法可以通过以下方式实现：服务器基于反馈信息和意图信息，选取至少一个工具，并将选取出的工具增加到工具执行计划中，得到修改后的工具执行计划。

具体的，当服务器判断出工具执行计划中存在缺失的工具和/或计划执行结果不完备时，可以获取 checker 模块提供的反馈信息以及用户的意图信息，并通过 Planner 模块再次从预定义的工具集合中选取至少一个工具增加到现有的工具执行计划中，以此生成修改后的工具执行计划。其中，大模型可以基于工具本身的功能以及应用场景的需要补充选取至少一个工具，以弥补当前工具执行计划的不足或者完善当前的计划执行结果。随后，Planner 模块可以将修改后的工具执行计划提交至 caller 模块进行执行，再由 checker 模块进行判断，以得到更加完整和准确的计划执行结果。

在本实施例中，再次选取的工具可以以增量的方式增加到原有的工具执行计划中，也可以替换掉原有的工具执行计划选取不正确的工具，还可以重复调用已经选取过的工具（如知识库搜索工具），以生成新的调用结果。此外，checker 模块提供的反馈信息可以包含工具执行计划中工具的检查结果和/或计划执行结果检查结果，以使得 Planner 模块可以基于反馈信息确定工具执行计划中缺失或者错误的工具，并补充新的工具，以此提高工具选取的效率和准确性。

上述实施方式通过基于 checker 模块的反馈信息和用户的意图信息，智能地选取并增加新的工具到工具执行计划中，可以使得系统能够针对不同的情况灵活地调整和优化工具执行计划，以确保信息的完备性和准确性，使得系统在面对复杂多变的对话场景时，能够保持较高的准确率和响应速度，从而减少人工干预的需要，降低系统的维护成本，提高系统的整体性能和用户体验。

在一个可选的实施方式中，计划执行结果可以通过设置在校验检查模块的大模型中的提示模板辅助检查，例如，大模型的提示模板中可以设置如下内容：

任务：你会接收到商家的问题、上下文信息以及小助手已经调用的工具和结果，并以此正确判断小助手是否已经尽力了。请始终记得小助手的工具列表（你也只能参考这些工具进行评判）：`{tool_hum_names}`。

任务要点：

- 1.理解问题：你需要仔细理解商家的问题，以便评估小助手的理解是否准确。
- 2.小助手能力边界：你一定要注意小助手的能力边界，如果无法通过现有工具获取解决方案，则认为小助手已经尽力了。
- 3.禁止工具重试：如果某工具调用失败，则没有重试的必要。

Question：用户的问题；Observation：调用的工具及工具结果；Thought：你一定要使用辩证思维围绕用户问题来思考，判断小助手是否已经尽力了，是否有问题相关的工具没有调用等；Check：Yes or No；小助手是否已经在能力范围内尽力了；Final Answer：当 Check 为 No 时，请在工具列表中选择正确的工具；当 Check 为 Yes 时回复“无”即可。

在上述实施方式中，通过在大模型的提示模板中设置计划执行结果的检查步骤、注意事项以及相关案例，可以使得计划执行结果的检查结果更加全面、准确、且没有遗漏。

步骤 206，服务器基于对话信息和上下文信息，对检查通过的计划执行结果进行整合，得到回复内容，并将回复内容发送至客户端。

具体的，在计划执行结果检查通过后，大模型可以基于检查通过的计划执行结果生成回复内容，其中，回复内容可以结合用户输入的对话信息生成，即在生成回复内容时，可以对计划执行结果中与对话信息相关的内容进行保留，而

将与对话信息无关的内容进行剔除，以此得到与对话信息较为相关的回复内容。在本实施例中，大模型还可以对回复内容进行其他形式的处理，如增加安抚性信息、增加链接信息等等，本实施例在此不做具体限定。其中，回复内容的生成过程可以通过E-Agent架构中的内容总结模块(Summary)来实现。

在一个可选的实施方式中，步骤206可以通过以下方式实现：服务器获取上下文信息，并基于对话信息和上下文信息，对计划执行结果中与对话信息相关的内容进行整合，得到回复内容。

具体的，服务器在进行内容整合时，首先可以获取当前的对话信息和上下文信息，然后按照一定的逻辑顺序，针对用户当前轮次发送的对话信息，对多个工具返回的结果以及文档块的内容进行组织，以确保信息的连贯性和准确性。此外，服务器还可以排除与当前轮次的对话信息不相关的噪声信息，以防止噪声信息干扰到最终答案的质量。最后，服务器可以将整合后的信息转化为自然语言，从而生成针对用户提问的回复内容。

上述实施方式通过针对用户的对话信息整合多个工具返回的结果和文档块的内容，生成针对对话信息的回复内容，能够为用户提供全面、连贯且准确的答复，从而能够有效的提升回复内容的准确性和针对性。此外，由于服务器能够基于给定的对话信息，在工具调用结果限定的范围内进行回答，因此不会额外引入其他信息，从而保证了回复内容的专业性。

在一个可选的实施方式中，回复内容可以通过以下至少一种方式生成：

206-1，服务器对计划执行结果中与对话信息不相关的内容进行删除，并对计划执行结果中保留的内容进行整合，得到回复内容。

具体的，服务器在进行内容整合时，可以通过自然语言处理技术对计划执行结果中的内容进行解析，以识别出与对话信息不相关的内容，并将其删除。然后，服务器可以对剩余的相关内容进行逻辑整合，以形成连贯、准确的自然语言回复内容。上述处理方式能够确保回复内容能够聚焦于用户的核心需求，避免引入不必要的噪声。

上述实施方式通过删除不相关内容，并整合剩余信息形成回复内容，能够有效的提高回复内容的针对性和准确性，减少因噪声干扰导致的误导用户的风险。同时，上述处理方式还可以提升回复内容的简洁性和可读性，使用户能够快速读取到所需的信息。

206-2，服务器基于对话信息，获取至少一个链接信息，并将链接信息添加到回复内容中。

具体的，在进行内容整合时，当服务器识别到对话信息中可以进一步添加参考信息或详细信息时，可以根据对话信息获取相关的链接信息（如图片链接、文档链接、网页链接等等），并将这些链接信息添加到回复内容中。在本实施方式中，链接信息可以是知识库中的文档链接，也可以是外部网站的链接。

上述实施方式通过添加链接信息到回复内容中，能够为用户提供更多的参考信息和详细信息，从而能够帮助用户更全面、更快速地了解相关问题的解决方案，以此提升用户的体验感和问题解决的时效性。

206-3，服务器在识别出对话信息中包含负面情绪时，基于对话信息生成安抚信息，并将安抚信息添加到回复内容中。

具体的，在进行内容整合时，当服务器识别到对话信息中包含负面情绪（如抱怨、不满等）时，服务器还可以基于对话信息生成安抚信息，如道歉信息、解释信息或建议信息等等，并将这些安抚信息添加到回复内容中。

上述实施方式通过针对用户发起的对话信息，在回复内容中添加安抚信息，能够在回复内容中尽量缓解用户的负面情绪，从而提升用户的体验感。

206-4，服务器在识别出对话信息中包含人工服务需求信息时，基于对话信息生成引导信息，并将引导信息添加到回复内容中。

具体的，在进行内容整合时，当服务器识别到对话信息中包含人工服务需求信息（如用户要求转人工服务等）时，服务器可以基于对话内容生成引导信息，如提供客服的联系方式、向用户描述任务型对话能够为用户提供的服务内容等等，并将这些引导信息添加到回复内容中，以引导用户快速解决问题。

上述实施方式通过添加引导信息到回复内容中，能够为用户提供更加有效的解决方案，减少用户的等待时间和沟通成本。同时，上述方式还能够提高服务器的响应速度和解决问题的能力，以增强用户的体验感。

206-5，服务器在识别出对话信息与预设领域不相符时，基于对话信息生成拒识信息，并将拒识信息作为回复内容。

具体的，在进行内容整合时，当服务器识别出对话信息与预设领域不相符时（如在商家服务领域中，用户提出与商家服务内容无关的问题），服务器可以基于对话内容生成拒识信息，并将拒识信息作为回复内容发送给用户。其中，拒识信息指的是表达无法处理用户提出的问题的相关信息，该信息可以基于用户当前轮次发起的对话信息生成。

上述实施方式通过在对话信息与当前领域不相符时，在回复内容中添加拒识信息，能够避免回答与预设领域不相符的问题，从而减少用户的混淆和误解。

206-6，服务器在预设的字数和语言风格的约束条件下，对计划执行结果中与对话信息相关的内容进行整合，得到结构化的回复内容。

具体的，在进行内容整合时，服务器可以在预设的字数和语言风格的约束条件下，对计划执行结果中与对话信息相关的内容进行整合和分段处理，从而得到结构化的回复内容。其中，结构化的回复内容可以包括按照主题划分段落的正文以及结论等各部分，并遵循一定的逻辑结构和语言风格进行排列。

上述实施方式通过按照一定的字数要求和语言风格生成结构化的回复内容，能够为用户提供清晰、易读的回复内容，从而帮助用户快速理解解决方案。

206-7，服务器基于意图信息，通过预设的信息检索模型，在回复内容经验库中搜索出至少一个案例，并将案例设置在提示模版中，以通过提示模版确定回复内容。

具体的，服务器可以根据对话信息对应的意图信息，通过预设的信息检索模型（如稀疏检索模型、稠密向量检索模型或多向量检索模型中的至少一种模型），在回复内容经验库中搜索出至少一个与当前意图相关的案例，然后将搜索到的案例设置在大模型的提示模版中，以通过提示模版确定最终的回复内容。

上述实施方式通过基于意图信息搜索相似案例，并基于搜索出的案例确定回复内容，能够为用户提供更加个性化和针对性的回复内容。同时，上述方式能够充分利用历史数据和经验库中的信息，提高回复内容的准确性和可靠性。

在一个可选的实施方式中，回复内容可以通过设置在内容总结模块的大模型中的提示模板辅助生成，例如，大模型的提示模板中可以设置如下内容：

任务目标：

- 1、对于商家提出的非外卖领域相关的问题，你需要委婉地拒绝回答。
- 2、对于商家的问题，你需要结合已知的信息和商家的情绪，根据这些信息做出人性化总结性的回答。
- 3、对于商家的问题，你需要把该问题结合到历史对话中一起进行判断。
- 4、引导咨询：在商家尝试联系客服经理或客服时，你需要积极引导他们直接向你提问，尽力处理并解决他们的问题。
- 5、输出的内容要分层分段，方便商家快速阅读。

ninput: <history_dialogue>: 商家与小助手的历史对话; <tools>: 当前调用的工具以及结果; <Links You Can Quote>: 你可以引用的链接列表; <Query>: 商家当前输入的内容;

output: Thought: 先从<Query>中判断商家是否有负面情绪，然后参考<history_dialogue>, <tools>, <Links You Can Quote>的内容; 进行总结性的回复;

Final Answer: 当商家有负面情绪时，先做出安抚; 然后给出对商家的回复。

注意：

*当提到XX号订单时，指的是订单流水号XX。

*当<tools>中的 Action 和 Observation 不为空时，你的 Thought 必须是合理且简洁的。

*从 Observation 的结果中只筛选出与商家当前问题相关的内容进行回答；

*<知识>中可能包含一些链接和图片，这些信息会以<链接-1>、<图片-1>的形式给出，这些也是重要的信息，在回答中直接保持<链接-1>、<图片-1>等原始格式进行引用；不要虚构<链接>，<link>，<知识>，图片等进行回答。

*Final Answer 要求简洁清晰，请严格从给定的 Observation 中筛选出合适且与商家问题相关的内容来回答问题，不能基于自身的知识回答，不要提供超出“Observation”的建议或做法。

案例：

ninput: <history_dialogue>: 空；<tools>: Action: knowledge_retrieval_tool, Observation: 外卖平台客服电话号码是 10105757。<Links You Can Quote>: []

<Query>: 人工呢？

noutput: Thought: 商家无负面情绪，就根据商家的问题进行回答，商家想寻找人工客服，我的责任是引导他们向我提问，所以我在提供电话号码的同时引导商家向我提问。Final Answer: 外卖客服电话号码是 10105757，如果您有任何在外卖平台上的问题，都欢迎向我提问，我会尽力帮您解答。

在一个可选的实施方式中，回复内容可以通过预训练的大模型生成，其中，上述大模型的预训练方法可以通过以下方式实现：服务器基于预设领域中多种预设的对话场景，生成多种类型的信息回复案例，然后基于多种类型的信息回复案例，对预设的大模型进行参数微调，得到预训练后的大模型，其中，预设的大模型可以为通用领域的大模型，也可以为通过预设领域的样本数据集继续预训练后得到的大模型。

具体的，服务器在进行模型训练时，可以根据预设领域内的多种预设的对话场景，构建多种类型的信息回复案例，其中，构建的案例中可以涵盖不同的用户意图、问题类型和回复风格，以确保大模型能够适应各种复杂的对话环境。随后，服务器可以利用这些构建好的信息回复案例，对预设的大模型进行参数微调。在本实施例中，大模型的基座模型可以是通用领域的大模型，也可以是基于预设领域样本数据集进行继续预训练后得到的垂域模型。在参数微调的过程中，服务器还可以利用优化算法调整模型的参数，以最小化预测回复与真实案例之间的误差，从而提高模型在预设领域内的回复准确性。

步骤 207，客户端接收并显示回复内容。

具体的，在生成回复内容之后，服务器可以将回复内容发送至客户端，客户端响应于接收到回复内容，可以对回复内容进行展示。例如，客户端可以通过即时通讯页面、搜索结果页面等多种页面对回复内容进行展示。在本实施例中，服务器在输出回复内容时，还可以输出其他信息，例如，服务器还可以输出用户的对话信息、意图信息、工具名称、工具调用结果、大模型的思考信息等等。在此场景下，服务器可以仅将大模型输出的部分信息发送至客户端进行展示，例如，服务器可以仅将回复内容发送至客户端，然后将其他信息记录在日志中，以便于日后进行分析处理。

在一个可选的实施方式中，客户端可以将回复内容的反馈信息发送至服务器。其中，反馈信息可以包括正反馈信息和负反馈信息。在本实施例中，正反馈信息和负反馈信息可以通过反馈控件的触发操作和/或用户对回复内容的评价内容生成。服务器在接收到客户端发送的反馈信息之后，可以基于反馈信息对经验库中的案例进行相应的操作，如新增案例，为案例增加经验分、减少案例的经验分、删除已有案例等等，以此提升经验库中案例的准确性和有效性。

步骤 208，服务器生成至少一种类型的案例，并根据案例的类型，将案例存储在对应的经验库中。

在本实施例中，服务器存储的案例可以包括意图识别案例、工具调用案例和信息回复案例中的至少一种类型的案例。其中，意图识别案例可以存储在意图信息经验库中，工具调用案例可以存储在执行计划经验库中，信息回复案例可以存储在回复内容经验库中。上述各种类型的案例可以在大模型训练之前、训练的过程中以及使用的过程中等各个阶段进行新增、修改和删除，以确保经验库中的案例能够尽量覆盖到预设领域中的多种对话场景，并能够解决用户在预设领域中面对的真实问题，以此提高问题解决的速度和质量。

在本实施例中，经验库中的案例可以包括两种类型，分别为结果直用型和样例型。其中，结果直用型指的是在用户提出的问题命中了某一类型的案例时，案例可以直接使用，此时，不需要调用大模型重新生成工具执行计划、回复

内容等结果，可以有效的降低计算资源；样例型指的是可以作为 few-shot（少量样例）放入大模型提示模版的上下文中，以指导大模型根据提示模版中的内容以及设置在提示模块中的样例，生成意图信息、工具执行计划、回复内容等内容。在本实施例中，在 E-Agent 的各环节，包括意图识别模块、计划制定模块和内容总结模块，均有各自独立的经验库，上述各经验库中均存储有对应的案例，这些案例可用于指导各个环节的大模型生成相对应的意图信息、工具执行计划、回复内容等内容，以提高工具执行计划或者回复内容的生成速度和质量。

在本实施例中，经验库中案例可以采用 K-V 结构进行存储。其中，V 指的是案例所在环节对应的直用型案例或样例型案例的具体内容，K 指的是意图提取环节中输出的单轮核心意图信息。例外的是，意图提取环节的经验 K 需要特殊处理，具体可以由最近至少一轮的用户的对话信息拼接而成。

进一步的，在任务型对话实施的各个环节中，如果匹配到经验库中的案例，则代表当前场景是熟悉并且有经验的场景，此时，可以使用经验数据（即案例）。在本实施例中，案例召回的过程可以认为是一个检索匹配任务，其核心在于如何高效和准确的判断当前情景是否精准匹配上数据库里的案例，具有性能和匹配精度要求极高的特点。基于此，本实施例采用 HybridRetrieval（搜索模型集成）的方式，通过组合稀疏检索（BM25）、稠密向量检索（BGE）和多向量检索（Multi-Vector）三种检索模型进行相结合的方式，进行信息检索和案例召回。其中，信息检索模型的架构如图 5 所示，信息检索模型的检索效果如图 6 所示，由图 6 可见，在选取阈值大于 0.79 时，hit@1（信息检索任务中的评价指标，指的是在给定的用户请求下，系统返回的第一个结果（top-1 result）是否命中（hit）了用户的目标或需求）的准确率可以稳定在 82% 以上，同时，召回率可以稳定在 93%，并且能够满足耗时控制在 500ms 内的性能要求。

进一步的，在案例生成方面，如图 4 所示，主要通过面向专家（即图 4 中的全过程反馈）和面向用户（即图 4 中的面向结果反馈）这两种方式生成并优化经验库中的案例。其中，在面向专家方面，可以在回复内容生成的过程中进行全程指导、并且可以实时进行改进和学习，以沉淀经验方案；在面向真实的用户方面，可以采用线上用户面向结果的反馈，以实时影响方案的经验数值。

在一个可选的实施方式中，案例可以通过以下至少一种方式生成：

208-1，接收用户与大模型交互时产生的意图信息、工作执行计划和回复内容，并基于用户对意图信息、工作执行计划和回复内容的第一正负反馈信息，对意图信息、工作执行计划和回复内容进行反馈操作，得到至少一个案例。

具体的，预设领域的用户（即专家）可以通过与基于 E-Agent 架构的任务型对话模型进行提问和交互的方式生成和优化案例，并且，专家能够在此过程中实时观测到 Agent 的中间过程以及最终回复。具体来说，在交互过程中，若 Agent 的回复或中间过程存在错误，专家可以即时指出问题所在并提供改进建议。然后，服务器可以根据专家的反馈，通过大模型或人工干预的方式生成相应的反馈信息。进一步的，专家可以针对错误的信息发起重试请求，在重试时，服务器可以将前一轮的反馈信息加入到大模型请求的 Prompt 中，并再次发起调用，然后重复此过程直至专家满意。最后，服务器可以从各个环节的大模型输入、输出和反馈信息中抽取出样例型和直用型经验，存储至经验库中。

上述实施方式通过预设领域的用户全程指导和实时反馈，能够精确识别并改进模型在回复内容生成过程中的问题，以确保生成的案例具有高质量和准确性。同时，通过不断积累专家经验，系统能够持续优化模型性能，提高模型在类似场景下的适应性。此外，通过抽取和存储样例型和直用型经验，系统能够快速响应新场景和新问题，并提升任务型对话的内容回复质量。

208-2，接收用户对回复内容的第二正负反馈信息，并基于第二正负反馈信息，对回复内容生成过程中产生的意图信息、工作执行计划和回复内容进行反馈操作，得到至少一个案例。

具体的，服务器还可以实时接收线上用户对于回复内容的正负反馈信息。具体来说，在线上用户与大模型交互的过程中，服务器可以实时接收线上用户对于回复内容的正负反馈信息。其中，用户可以通过点踩、点赞、发送评论等多种方式生成回复内容的正负反馈信息，以使得服务器可以根据用户做出的反馈信息对相应案例进行正负反馈操作，如增减案例，或增减案例的经验分等。这种基于用户实时反馈的方式，能够迅速捕捉用户需求，并不断优化案例库。

上述实施方式通过实时接收和处理用户的正负反馈信息，能够迅速识别出用户需求，并据此优化和更新案例库。这种方式能够显著降低错误经验对于用户体验的负面影响，同时能够提升用户的满意度。此外，通过收集用户对于回复内容的正负反馈信息的方式，服务器能够自动识别并存储有价值的用户体验，从而能够进一步丰富和完善案例库，以此提高任务型对话的服务质量。

在一个可选的实施方式中，参照图 4 中以面向过程反馈的方式生成和优化案例的过程，步骤 208-1 可以通过以下至少一种方式实现：

208-1-1，服务器在接收到用户对意图信息、工作执行计划和回复内容为正反馈信息时，对正反馈信息对应的意图信息、工作执行计划和回复内容对应的案例进行正反馈操作，其中，正反馈操作包括增加案例的经验分，和/或，增加新的案例。

具体的，当服务器接收到用户对意图信息、工作执行计划和回复内容为正反馈信息时，可以首先识别并定位到正反馈信息对应的案例。然后，系统可以对该案例进行正反馈操作，具体操作包括但不限于增加该案例的经验分，以便在后续相似问题中提高该案例的优先级。同时，服务器还可以根据正反馈信息的具体内容，将相似或相关的正反馈案例进行归并或整合，以生成新的案例。

上述实施方式通过收集用户对案例的正反馈操作，能够迅速识别并强化用户满意的案例，从而提高案例库的质量和可用性。同时，新生成的案例能够进一步丰富案例库的内容，以提高系统的适应性和灵活性。

208-1-2，服务器在接收到用户对意图信息、工作执行计划和回复内容为负反馈信息时，对负反馈信息对应的意图信息、工作执行计划和回复内容进行纠错处理，生成纠错信息，并将纠错信息设置在提示模版中，以通过提示模版重新生成意图信息、工作执行计划和回复内容。

具体的，当服务器接收到用户对意图信息、工作执行计划和回复内容为负反馈信息时，可以首先分析负反馈的具体原因和内容，并定位到出错的案例。然后，可以对出错的案例进行纠错处理，并生成纠错信息。这些纠错信息可以包括错误信息的具体描述、正确的处理方式或建议的改进方向等等。然后，服务器可以将纠错信息设置在大模型的提示模版中，该模版可用于指导后续的案例生成。最后，大模型可以根据提示模版重新生成意图信息、工作执行计划和回复内容，以确保新的内容能够避免之前的错误，以提高用户的满意度。

在一个可选的实施方式中，步骤 208-1-2 中通过提示模版重新生成意图信息、工作执行计划和回复内容的方法可以通过以下方式实现：服务器在识别出通过案例生成的意图信息、工作执行计划和回复内容对应的案例有误时，在重新生成意图信息、工作执行计划和回复内容时，更换有误的案例，并对有误的案例进行负反馈操作，其中，负反馈操作包括减少案例的经验分和/或删除案例。

具体的，当服务器识别出通过案例生成的意图信息、工作执行计划和回复内容存在错误时，可以首先识别出出错的案例，并确定该案例在生成意图信息、工作执行计划和回复内容中的哪个环节出现了错误。然后，服务器可以更换有误的案例，并基于其他正确或更合适的案例，重新进行意图识别、工作执行计划的制定和回复内容的生成。在重新生成的过程中，服务器还可以对出错案例进行负反馈操作。这些负反馈操作包括但不限于减少该案例的经验分，以降低其在未来类似情况下的优先级；在特定情况下，如果案例的错误严重或多次出现类似错误，服务器还可以选择删除该案例，以避免该案例影响到系统的性能。

在一个可选的实施方式中，参照图 4 中以面向结果反馈的方式生成和优化案例的过程，步骤 208-2 可以通过以下至少一种方式实现：

208-2-1，服务器在接收到用户对回复内容为正反馈信息时，对通过案例生成的意图信息、工作执行计划和回复内容对应的案例进行正反馈操作，其中，正反馈操作包括增加案例的经验分。

具体的，当服务器接收到用户对于回复内容为正反馈信息时，可以定位到回复内容对应的案例中，即定位到通过案例生成的意图信息、工作执行计划和回复内容对应的案例中，然后，服务器可以对这些案例进行正反馈操作，主要

包括增加案例的经验分。其中，增加案例的经验分可以提高案例在未来相似场景下的优先级，使得系统更倾向于使用经验分更高的案例来生成回复内容。

208-2-2，服务器在接收到用户对回复内容为正反馈信息时，将未通过案例生成的意图信息、工作执行计划和回复内容增加为新的案例。

具体的，当服务器接收到用户对于回复内容为正反馈信息时，如果回复内容、工具执行计划和回复内容并不是通过案例生成的，而是通过大模型基于意图信息重新生成的，则服务器可以将新生成的意图信息、工作执行计划和回复内容作为一个全新的案例添加到案例库中。通过这种方式，系统可以不断学习和积累新的知识和经验，以应对更加多样化和复杂化的用户需求。

208-2-3，服务器在接收到用户对回复内容为负反馈信息时，对通过案例生成的意图信息、工作执行计划和回复内容对应的案例进行负反馈操作，其中，负反馈操作包括减少案例的经验分和/或删除案例。

具体的，当服务器接收到用户对于回复内容为负反馈信息时，可以定位到对应的案例，然后对这些案例进行负反馈操作，主要包括减少案例的经验分，另外，在某些情况下，也可以删除这些案例。其中，减少案例的经验分可以降低这些案例在未来被选中的概率，而删除案例则直接移除了潜在的错误或低质量的案例，从而有助于提高系统的整体性能和准确性。

208-2-4，服务器在接收到用户对回复内容为负反馈信息时，对未通过案例生成的意图信息、工作执行计划和回复内容进行删除。

具体的，当服务器接收到用户对于回复内容为负反馈信息时，如果该回复内容对应的意图信息、工具执行计划和回复内容并不是通过案例生成的，则服务器可以直接删除掉这些意图信息、工作执行计划和回复内容，从而有助于保持案例库的清洁性和高效性，避免存储无效或低质量的内容。

上述实施方式通过根据用户的实时反馈来动态地优化和调整经验库中的案例，可以提高系统的性能以及用户体验。首先，上述方式通过正反馈操作增加案例的经验分，能够更准确地识别和使用高质量的案例来生成回复，从而提高回复的准确性和可靠性。同时，负反馈操作能够减少或删除低质量的案例，以降低错误回复的概率。而且，通过添加新的案例和删除无效案例，系统能够不断学习和积累新的知识和经验，以应对更加多样化和复杂的用户需求，使得系统具有更强的适应性和灵活性，从而能够更好地满足用户的个性化需求。此外，通过优化案例库，系统能够生成更加准确和有用的回复，从而提升用户的体验感。同时，用户的反馈也能够被充分利用，增强了用户与系统之间的互动性。

需要说明的是，本申请所涉及的用户信息（包括但不限于用户设备信息、用户个人信息等）和数据（包括但不限于用于分析的数据、存储的数据、展示的数据等），均为经用户授权或者经过各方充分授权的信息和数据。另外，上述实施例中的各个步骤所对应的标号仅起到标识作用，并不作为对于步骤执行顺序的限定，各个实施例中的步骤执行顺序可以根据实际情况进行设置。

进一步的，作为图 1 至图 6 所示方法的具体实现，本申请实施例提供了一种任务型对话装置，如图 7 所示，该装置包括：

对话信息发送模块 31，可用于响应于接收到用户发起的对话信息，发送所述对话信息；

回复内容接收模块 32，可用于接收并显示所述回复内容，其中，所述回复内容的生成过程包括：基于所述对话信息，通过大模型确定所述用户的意图信息，并根据所述意图信息生成工具执行计划，执行所述工具执行计划，得到计划执行结果，对所述计划执行结果进行检查，并基于检查通过的计划执行结果生成所述回复内容。

进一步的，作为图 1 至图 6 所示方法的具体实现，本申请实施例提供了一种任务型对话装置，如图 8 所示，该装置包括：

意图识别模块 41，可用于响应于接收到用户发起的对话信息，通过大模型确定所述用户的意图信息；

计划制定模块 42，可用于根据所述意图信息生成工具执行计划；

调用执行模块 43，可用于执行所述工具执行计划，得到计划执行结果；

校验检查模块 44，可用于对所述计划执行结果进行检查；

内容总结模块 45，可用于基于检查通过的计划执行结果生成回复内容。

需要说明的是，本申请实施例提供的一种任务型对话装置所涉及各功能单元的其他相应描述，可以参考图 1 至图 6 方法中的对应描述，在此不再赘述。

本申请实施例还提供了一种计算机设备，具体可以为个人计算机、服务器、网络设备等，该计算机设备包括总线、处理器、存储器和通信接口，还可以包括输入输出接口和显示设备。其中，该计算机设备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和内存存储器。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位置信息。该计算机设备的网络接口用于与外部的终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各方法实施例中的步骤。

本领域技术人员可以理解，上述的计算机设备的结构，仅仅是与本申请方案相关的部分结构，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，具体的计算机设备可以包括更多或更少的部件，或者组合某些部件，或者具有不同的部件布置。

在一个实施例中，提供了一种计算机可读存储介质，所述计算机可读存储介质可以是非易失性，也可以是易失性，其上存储有计算机程序，该计算机程序被处理器执行时实现上述各方法实施例中的步骤。

在一个实施例中，提供了一种计算机程序产品，包括计算机程序，该计算机程序被处理器执行时实现上述各方法实施例中的步骤。

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，所述的计算机程序可存储于一非易失性计算机可读存储介质中，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，本申请所提供的各实施例中所使用的对存储器、数据库或其它介质的任何引用，均可包括非易失性和易失性存储器中的至少一种。非易失性存储器可包括只读存储器（Read-Only Memory, ROM）、磁带、软盘、闪存、光存储器、高密度嵌入式非易失性存储器、阻变存储器（ReRAM）、磁变存储器（Magnetoresistive Random Access Memory, MRAM）、铁电存储器（Ferroelectric Random Access Memory, FRAM）、相变存储器（Phase Change Memory, PCM）、石墨烯存储器等。易失性存储器可包括随机存取存储器（Random Access Memory, RAM）或外部高速缓冲存储器等。作为说明而非局限，RAM 可以是多种形式，比如静态随机存取存储器（Static Random Access Memory, SRAM）或动态随机存取存储器（Dynamic Random Access Memory, DRAM）等。本申请所提供的各实施例中所涉及的数据库可包括关系型数据库和非关系型数据库中至少一种。非关系型数据库可包括基于区块链的分布式数据库等，不限于此。本申请所提供的各实施例中所涉及的处理器可为通用处理器、图形处理器、数字信号处理器、可编程逻辑器、基于量子计算的数据处理逻辑器等，不限于此。

以上实施例的各技术特征可以进行任意的组合，为使描述简洁，未对上述实施例中的各个技术特征所有可能的组合都进行描述，然而，只要这些技术特征的组合不存在矛盾，都应当认为是本说明书记载的范围。

以上所述实施例仅表达了本申请的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对本申请专利范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本申请构思的前提下，还可以做出若干变形和改进，这些都属于本申请的保护范围。因此，本申请的保护范围应以所附权利要求为准。

权 利 要 求 书

- 1、一种任务型对话方法，其特征在于，所述方法包括：
- 客户端响应于接收到用户发起的对话信息，将所述对话信息发送至服务器；
- 5 所述服务器响应于接收到所述对话信息，通过大模型确定所述用户的意图信息，并根据所述意图信息生成工具执行计划，执行所述工具执行计划，得到计划执行结果，对所述计划执行结果进行检查，并基于检查通过的计划执行结果生成回复内容，将所述回复内容发送至所述客户端；
- 所述客户端接收并显示所述回复内容。
- 2、根据权利要求1所述的方法，其特征在于，所述服务器响应于接收到所述对话信息，通过大模型
- 10 确定所述用户的意图信息，包括：
- 所述服务器响应于接收到所述对话信息，获取所述对话信息的上下文信息和预设的应用场景信息；
- 所述服务器将所述应用场景信息作为约束条件，基于所述对话信息和所述上下文信息，确定所述用户的意图信息。
- 3、根据权利要求2所述的方法，其特征在于，所述服务器将所述应用场景信息作为约束条件，基于
- 15 所述对话信息和所述上下文信息，确定所述用户的意图信息，包括：
- 所述服务器在所述应用场景信息约束的话题范围内，对所述对话信息和所述上下文信息进行改写，得到所述用户的至少一条意图信息；和/或，
- 所述服务器对所述对话信息和所述上下文信息中与所述应用场景信息相关的信息进行保留，并根据保留的内容的主题生成至少一条意图信息；和/或，
- 20 所述服务器将所述对话信息和所述上下文信息中的时间信息和数字信息保留在所述用户的意图信息中；和/或，
- 所述服务器将所述对话信息和所述上下文信息中的指代信息替换为名称信息保留在所述用户的意图信息中；和/或，
- 所述服务器在识别出所述上下文信息中包含与所述对话信息不相同的差异性话题时，将与所述差异性话题相关的内容在所述意图信息中删除；和/或，
- 25 所述服务器在判断出所述对话信息和所述上下文信息未在所述应用场景信息约束的话题范围内时，将所述对话信息作为所述用户的意图信息；和/或，
- 所述服务器基于所述对话信息和所述上下文信息，通过预设的信息检索模型，在意图信息经验库中搜索出至少一个案例，并将所述案例设置在提示模版中，以通过所述提示模版确定所述用户的意图信息，其中，所述信息检索模型包括稀疏检索模型、稠密向量检索模型和多向量检索模型中的至少一种模型。
- 30 4、根据权利要求1至3任一项所述的方法，其特征在于，所述意图信息通过预训练的大模型生成，所述大模型的预训练方法包括：
- 所述服务器基于预设领域的样本数据集，对通用领域的大模型进行继续预训练，其中，所述样本数据集中设置了随机噪声，且不包含样本用户信息；
- 35 所述服务器基于所述预设领域的意图识别案例，对继续预训练后的大模型进行参数微调；
- 所述服务器基于微调后的大模型，生成所述预设领域中预设问题的意图识别信息，并基于所述意图识别信息的正负反馈信息，通过强化学习算法迭代更新所述大模型的参数，得到预训练后的大模型。
- 5、根据权利要求1所述的方法，其特征在于，所述服务器根据所述意图信息生成工具执行计划，包括：

所述服务器获取上下文信息，并根据所述上下文信息在工具库中查找与所述意图信息相关的多个候选工具，生成候选工具列表；

所述服务器基于所述意图信息，在所述候选工具列表中选择出至少一个工具，并对选取的工具进行排列组合，得到所述工具执行计划。

5 6、根据权利要求5所述的方法，其特征在于，所述服务器基于所述意图信息，在所述候选工具列表中选择出至少一个工具，包括：

所述服务器基于所述意图信息，在所述候选工具列表中进行关键词匹配，筛选出至少一个工具；和/或，

10 所述服务器在判断出当前轮次的意图信息与上一轮次的意图信息含义相同时，将与上一轮次的工具不同的工具作为当前轮次选取的工具；和/或，

所述服务器将与选取的工具互斥的工具作为当前轮次禁用的工具；和/或，

所述服务器在基于所述意图信息未选取工具时，将知识检索工具作为当前轮次选取的工具；和/或，

15 所述服务器基于所述意图信息，通过预设的信息检索模型，在执行计划经验库中搜索出至少一个案例，并基于所述案例在所述候选工具列表中选择出至少一个工具，其中，所述信息检索模型包括稀疏检索模型、稠密向量检索模型和多向量检索模型中的至少一种模型；

其中，所述服务器基于所述案例在所述候选工具列表中选择出至少一个工具，包括：

所述服务器在所述案例与所述意图信息的匹配度大于预设阈值时，将所述案例中的工具作为选取的工具；和/或，

20 所述服务器在所述案例与所述意图信息的匹配度小于或等于所述预设阈值时，将所述案例设置在提示模版中，以通过所述提示模版，在所述候选工具列表中选择出至少一个工具。

7、根据权利要求5所述的方法，其特征在于，所述服务器对选取的工具进行排列组合，得到所述工具执行计划，包括：

所述服务器基于选取的各工具的工具参数之间的依赖关系，对所述工具进行顺序排列和/或并行排列；

25 所述服务器通过预设的运算符，对顺序排列和/或并行排列的工具进行运算连接，得到所述工具执行计划。

8、根据权利要求5至7任一项所述的方法，其特征在于，所述工具执行计划通过预训练的大模型生成，所述大模型的预训练方法包括：

所述服务器基于预设领域的样本数据集，对通用领域的大模型进行继续预训练；

所述服务器基于所述预设领域的工具调用案例，对继续预训练后的大模型进行参数微调；

30 所述服务器将微调后的大模型的参数权重从浮点数转换为预设位数的整数，得到预训练后的大模型。

9、根据权利要求1所述的方法，其特征在于，所述服务器执行所述工具执行计划，得到计划执行结果，包括：

所述服务器获取上下文信息，并在所述对话信息和所述上下文信息中提取出所述工具执行计划中的工具所需的工具参数；

35 所述服务器基于所述工具参数，按照所述工具执行计划中工具的排列组合的顺序，对所述工具进行串行调用和/或并行调用，得到所述计划执行结果；

其中，所述服务器基于所述工具参数，按照所述工具执行计划中工具的排列组合的顺序，对所述工具进行串行调用和/或并行调用，得到所述计划执行结果，包括：

所述服务器基于所述工具参数，对所述工具执行计划中顺序排列的工具进行串行调用，对并行排列的工具进行并行调用，得到所述计划执行结果。

10、根据权利要求9所述的方法，其特征在于，所述方法还包括：

所述服务器在识别出存在缺失的工具参数时，将缺失工具参数的工具的执行顺序后置，在已执行的
5 工具的执行结果中提取所述缺失的工具参数；和/或，

所述服务器基于缺失的工具参数，发起询问信息，并在所述用户对所述询问信息的答复信息中提取所述缺失的工具参数；和/或，

所述服务器在所述工具参数中提取出时间参数，并将所述时间参数中的相对时间修改为绝对时间；
和/或，

10 所述服务器在所述工具参数中提取出订单参数，并将所述订单参数中的用户表述修改为预设领域的标准表述；和/或，

所述工具执行计划中的工具包括知识库搜索工具、信息查询工具、功能设置工具、信息解析工具中的至少一种工具。

11、根据权利要求1所述的方法，其特征在于，所述服务器对所述计划执行结果进行检查，包括：

15 所述服务器获取上下文信息，并基于所述对话信息和所述上下文信息，对所述工具执行计划中的工具和所述计划执行结果进行检查；

所述服务器在判断出所述工具执行计划中不存在缺失的工具，和/或，所述计划执行结果完备时，确定所述计划执行结果检查通过；和/或，

20 所述服务器在判断出所述工具执行计划中存在缺失的工具，和/或，所述计划执行结果不完备时，循环执行以下过程，直至修改后的计划执行结果检查通过，和/或，所述计划执行结果的检查次数达到预设阈值：所述服务器基于所述工具执行计划和所述计划执行结果的检查结果生成反馈信息，并基于所述反馈信息和所述意图信息修改所述工具执行计划，执行修改后的工具执行计划，得到修改后的计划执行结果；

25 其中，所述服务器基于所述反馈信息和所述意图信息修改所述工具执行计划，包括：所述服务器基于所述反馈信息和所述意图信息，选取至少一个工具，并将选取出的工具增加到所述工具执行计划中，得到修改后的工具执行计划。

12、根据权利要求1所述的方法，其特征在于，所述服务器基于检查通过的计划执行结果生成回复内容，包括：

30 所述服务器获取上下文信息，并基于所述对话信息和所述上下文信息，对所述计划执行结果中与所述对话信息相关的内容进行整合，得到所述回复内容；

其中，所述服务器基于所述对话信息和所述上下文信息，对所述计划执行结果中与所述对话信息相关的内容进行整合，得到所述回复内容，包括：

所述服务器对所述计划执行结果中与所述对话信息不相关的内容进行删除，对所述计划执行结果中保留的内容进行整合，得到所述回复内容；和/或，

35 所述服务器基于所述对话信息，获取至少一个链接信息，并将所述链接信息添加到所述回复内容中；和/或，

所述服务器在识别出所述对话信息中包含负面情绪时，基于所述对话信息生成安抚信息，并将所述安抚信息添加到所述回复内容中；和/或，

所述服务器在识别出所述对话信息中包含人工服务需求信息时，基于所述对话信息生成引导信息，并将所述引导信息添加到所述回复内容中；和/或，

所述服务器在识别出所述对话信息与预设领域不相符时，基于所述对话信息生成拒识信息，并将所述拒识信息作为所述回复内容；和/或，

5 所述服务器在预设的字数和语言风格的约束条件下，对所述计划执行结果中与所述对话信息相关的内容进行整合，生成结构化的所述回复内容；和/或，

所述服务器基于所述意图信息，通过预设的信息检索模型，在回复内容经验库中搜索出至少一个案例，并将所述案例设置在提示模版中，以通过所述提示模版确定所述回复内容，其中，所述信息检索模型包括稀疏检索模型、稠密向量检索模型和多向量检索模型中的至少一种模型。

10 13、根据权利要求1或12所述的方法，其特征在于，所述回复内容通过预训练的大模型生成，所述大模型的预训练方法包括：

所述服务器基于预设领域中多种预设的对话场景，生成多种类型的信息回复案例；

所述服务器基于所述多种类型的信息回复案例，对预设的大模型进行参数微调，得到预训练后的大模型，其中，所述预设的大模型为通用领域的大模型，或通过预设领域的样本数据集继续预训练后得到

15 的大模型。

14、根据权利要求1所述的方法，其特征在于，所述方法还包括：

所述服务器生成至少一种类型的案例，并根据所述案例的类型，将所述案例存储在对应的经验库中，其中，所述案例包括意图识别案例、工具调用案例和信息回复案例中的至少一种案例；

其中，所述至少一种类型的案例的生成方法包括：

20 接收用户与大模型交互时产生的意图信息、工作执行计划和回复内容，基于所述用户对所述意图信息、所述工作执行计划和所述回复内容的第一正负反馈信息，对所述意图信息、所述工作执行计划和所述回复内容进行反馈操作，得到所述至少一个案例；和/或，

接收用户对所述回复内容的第二正负反馈信息，并基于所述第二正负反馈信息，对所述回复内容生成过程中产生的意图信息、工作执行计划和回复内容进行反馈操作，得到所述至少一个案例。

25 15、根据权利要求14所述的方法，其特征在于，所述服务器基于所述第一正负反馈信息，对所述意图信息、所述工作执行计划和所述回复内容进行反馈操作，得到至少一个案例，包括：

所述服务器在接收到用户对所述意图信息、所述工作执行计划和所述回复内容为正反馈信息时，对所述正反馈信息对应的意图信息、工作执行计划和回复内容对应的案例进行正反馈操作，其中，所述正反馈操作包括增加所述案例的经验分，和/或，增加新的案例；和/或，

30 所述服务器在接收到用户对所述意图信息、所述工作执行计划和所述回复内容为负反馈信息时，对所述负反馈信息对应的意图信息、工作执行计划和回复内容进行纠错处理，生成纠错信息，并将所述纠错信息设置在提示模版中，以通过所述提示模版重新生成意图信息、工作执行计划和回复内容；

其中，所述服务器通过所述提示模版重新生成意图信息、工作执行计划和回复内容，包括：

所述服务器在识别出通过案例生成的意图信息、工作执行计划和回复内容对应的案例有误时，在重

35 新生成意图信息、工作执行计划和回复内容时，更换有误的案例，并对所述有误的案例进行负反馈操作，其中，所述负反馈操作包括减少所述案例的经验分和/或删除所述案例。

16、根据权利要求14所述的方法，其特征在于，所述服务器基于所述第二正负反馈信息，对所述回复内容生成过程中产生的意图信息、工作执行计划和回复内容进行反馈操作，得到至少一个案例，包括：

所述服务器在接收到用户对所述回复内容为正反馈信息时，对通过案例生成的意图信息、工作执行计划和回复内容对应的案例进行正反馈操作，其中，所述正反馈操作包括增加所述案例的经验分；和/或，

所述服务器在接收到用户对所述回复内容为正反馈信息时，将未通过案例生成的意图信息、工作执行计划和回复内容增加为新的案例；和/或，

- 5 所述服务器在接收到用户对所述回复内容为负反馈信息时，对通过案例生成的意图信息、工作执行计划和回复内容对应的案例进行负反馈操作，其中，所述负反馈操作包括减少所述案例的经验分和/或删除所述案例；和/或，

所述服务器在接收到用户对所述回复内容为负反馈信息时，对未通过案例生成的意图信息、工作执行计划和回复内容进行删除。

- 10 17、一种任务型对话方法，其特征在于，所述方法包括：

响应于接收到用户发起的对话信息，发送所述对话信息；

接收并显示回复内容，其中，所述回复内容的生成过程包括：基于所述对话信息，通过大模型确定所述用户的意图信息，并根据所述意图信息生成工具执行计划，执行所述工具执行计划，得到计划执行结果，对所述计划执行结果进行检查，并基于检查通过的计划执行结果生成所述回复内容。

- 15 18、一种任务型对话方法，其特征在于，所述方法包括：

响应于接收到用户发起的对话信息，通过大模型确定所述用户的意图信息；

根据所述意图信息生成工具执行计划；

执行所述工具执行计划，得到计划执行结果；

对所述计划执行结果进行检查；

- 20 基于检查通过的计划执行结果生成回复内容。

19、一种存储介质，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18中任一项所述的方法。

20、一种计算机设备，包括存储介质、处理器及存储在存储介质上并可在处理器上运行的计算机程序，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18中任一项所述的方法。