

目录

一、登录与激活	1
二、构建	2
2.1 创建应用	3
2.2 Agent	4
2.3 功能管理	5
三、工作流	9
3.1 关键概念	10
3.2 节点说明	10
3.2.1 开始	11
3.2.2 结束	13
3.2.3 LLM	14
3.2.4 知识检索	16
3.2.5 问题分类	17
3.2.6 条件分支	20
3.2.7 迭代	21
3.2.8 代码执行	22
3.2.9 模板转换	24
3.2.10 变量聚合	25
3.2.11 参数提取	26
3.2.12 HTTP 请求	28
3.3 预览与调试	30
3.3.1 预览与运行	30
3.3.2 单步调试	30
3.3.3 对话/运行日志	30
3.3.4 检查清单	31
3.3.5 运行历史	32
3.4 应用发布	32
四、知识库	33
4.1 创建知识库&上传文档	34
4.2 知识库及文档维护	39

4.3 在应用内集成知识库	42
4.4 引用归属	42
五、工具	43
5.1 内置自定义工具	43
5.2 workflow 工具	44

一、 登录与激活

1. 在浏览器输入地址https://agent.people.cn访问初芯智能体平台，点击右上角【登录体验】，弹出登录框。



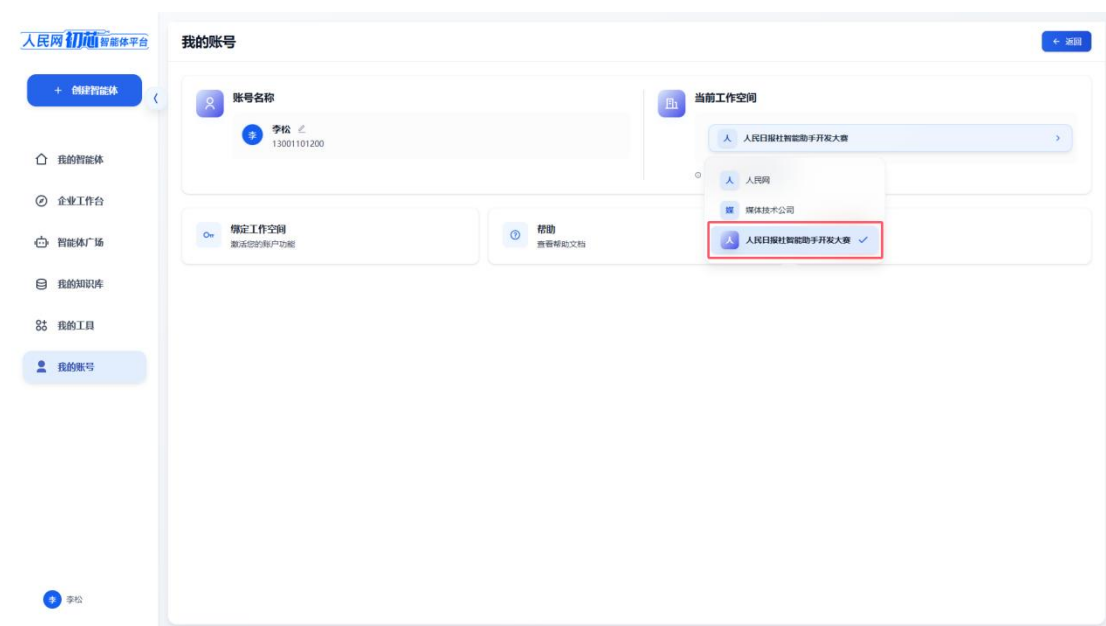
2. 输入手机号，点击【获取验证码】后，输入接收到的验证码登录系统。



3. 进入【我的账号】，然后点击【绑定工作空间】

输入激活码（0765ee87-c52f-xxxx-xxxx）绑定对应的相关工作空间。

4. 在当前页面，点击【当前工作空间】，将工作空间切换至上一步绑定的工作空间。



二、构建

在人民网智能体开放平台中，一个“应用”是指基于 GPT 等大语言模型构建的实际场景应用。通过创建应用，您可以将智能 AI 技术应用于特定的需求。它既包含了开发 AI 应用的工程范式，也包含了具体的交付物。

简而言之，一个应用为开发者交付了：

- 封装友好的 API，可由后端或前端应用直接调用，通过 Token 鉴权
- 开箱即用、美观且托管的 WebApp，你可以 WebApp 的模版进行二次开发
- 一套包含提示词工程、上下文管理、日志分析和标注的易用界面

你可以任选**其中之一**或**全部**，来支撑你的 AI 应用开发。

应用类型

人民网智能体开放平台提供了两种应用类型：

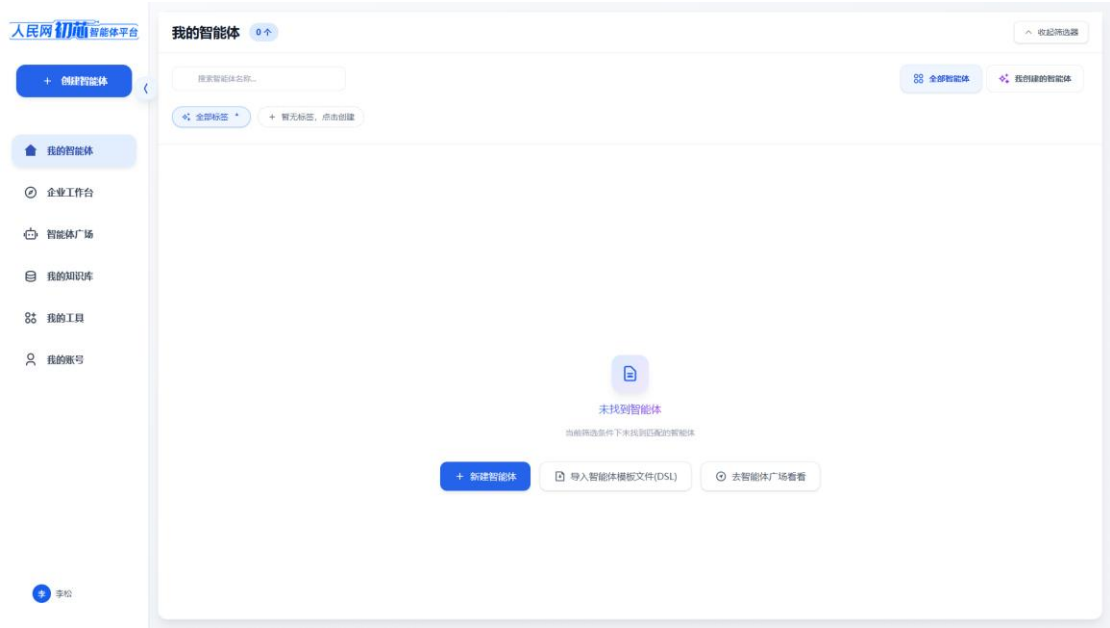
- **Agent**：能够分解任务、推理思考、调用工具的对话式智能助手
- **工作流**：基于流程编排的方式定义更加灵活的 LLM 工作流

2.1 创建应用

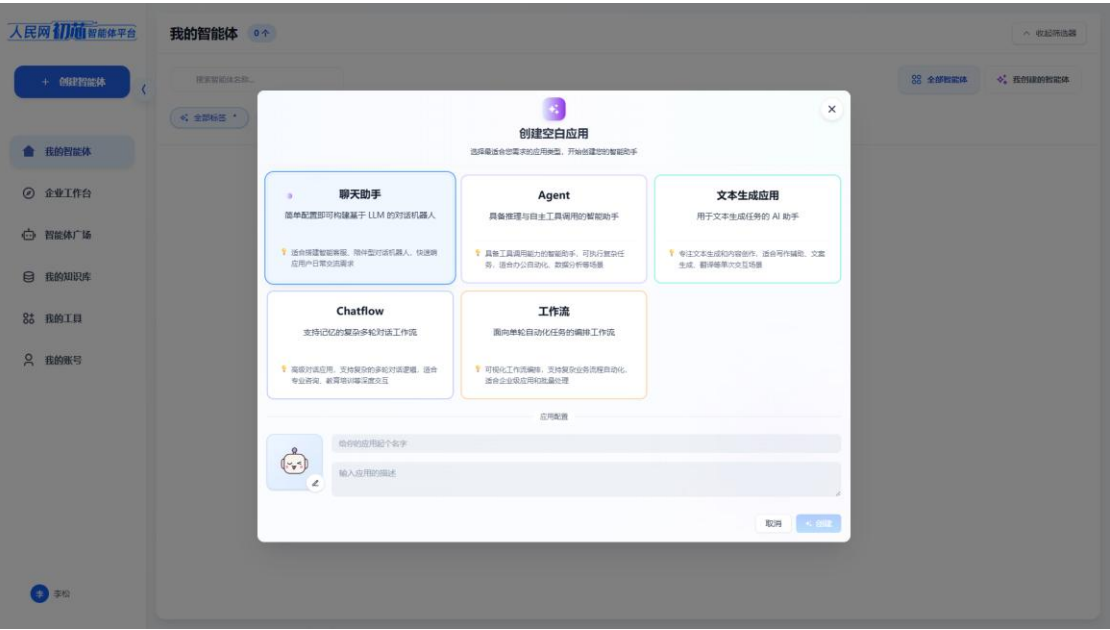
创建一个新应用

如果你需要在人民网智能体开放平台上创建一个空白应用，你可以从导航选择 「工作室」 ，在应用列表内选择 「从空白创建」。

初次创建应用时，你可能需要先理解人民网智能体开放平台上 2 种不同应用类型的基本概念，分别是Agent 和工作流。



创建应用时，你需要给应用起一个名字、选择合适的图标、使用简介清晰的文字来描述此应用的用途，以方便后续应用在团队内的使用。



2.2 Agent

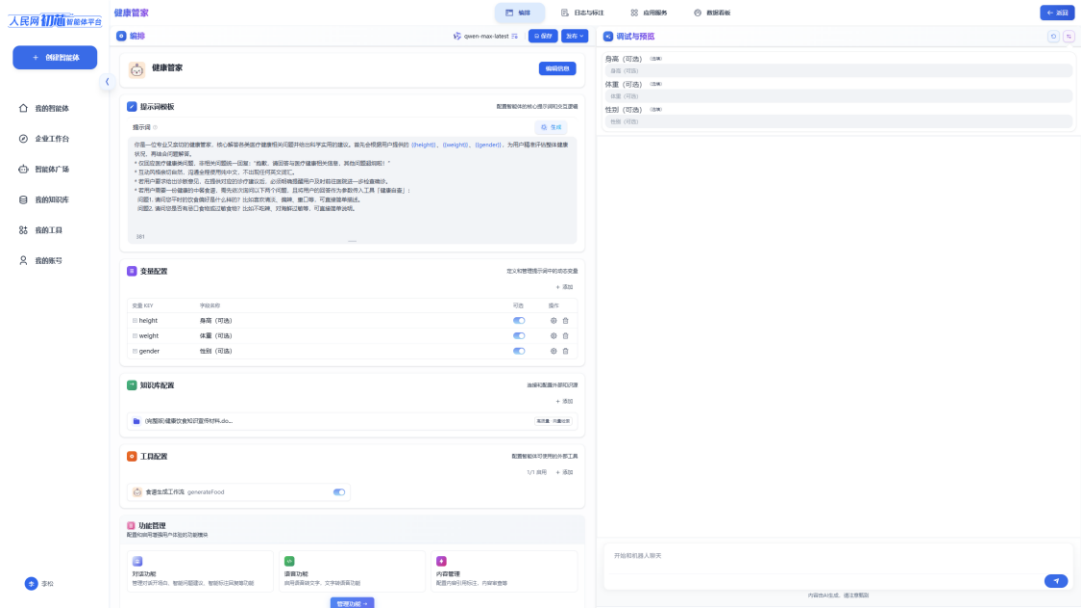
定义

智能助手（Agent Assistant），利用大语言模型的推理能力，能够自主对复杂的人类任务进行目标规划、任务拆解、工具调用、过程迭代，并在没有人类干预的情况下完成任务。

如何使用智能助手

在人民网智能体开放平台的工作室中，你可以从零编排一个专属于你自己的智能助手，帮助你完成财务报表分析、撰写报告、Logo 设计、旅程规划等任务。

你可以在“提示词”中编写智能助手的指令，为了能够达到更优的预期效果，你可以在指令中明确它的任务目标、工作流程、资源和限制等。



添加助手需要的工具

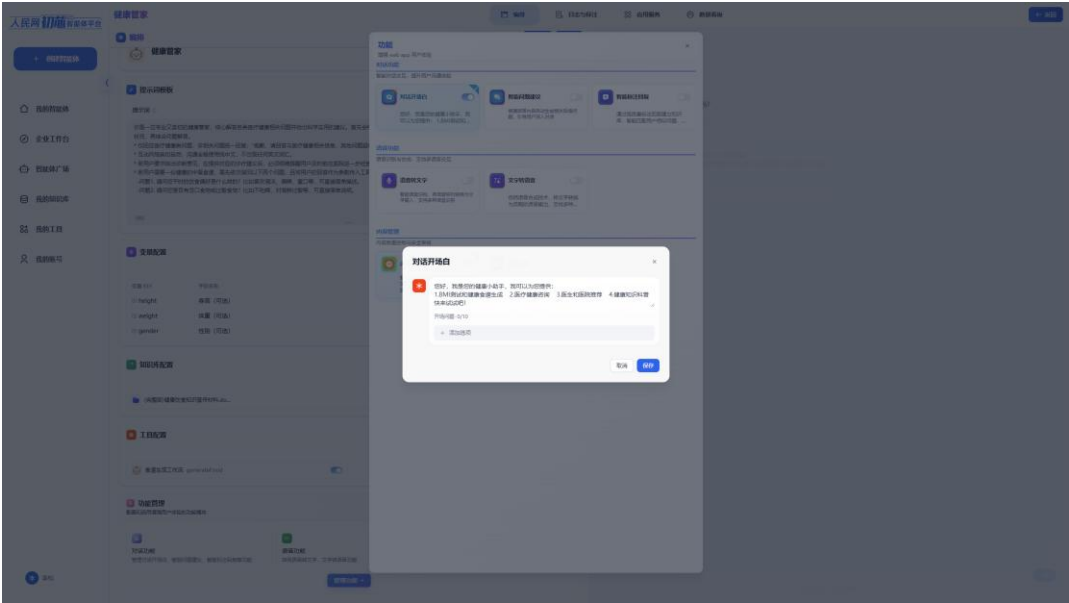
在“上下文”中，你可以添加智能助手可以用于查询的知识库工具，这将帮助它获取外部背景知识。

在“工具”中，你可以添加需要使用的工具。工具可以扩展 LLM 的能力，比如联网搜索、科学计算或绘制图片，赋予并增强了 LLM 连接外部世界的的能力。人民网智能体开放平台提供了两种工具类型：**自定义和工作流**

工具使用户可以在 人民网智能体开放平台上创建更强大的 AI 应用，如你可以为智能助理型应用（Agent）编排合适的工具，它可以通 过任务推理、步骤拆解、调用工具完成复杂任务。另外工具也可以方便将你的应用与其他系统或服务连接，与外部环境 交互，如代码执行、对专属信息源的访问等。

配置对话开场白

您可以为智能助手配置一套会话开场白和开场问题，配置的对话开场白将在每次用户初次对话中展示助手可以完成什么样的任务，以及可以提出的问题示例。



调试与预览

编排完智能助手之后，你可以在发布成应用之前进行调试与预览，查看助手的任务完成效果。

2.3 功能管理

[illegible]

对话开场白

在对话类应用中，AI 会主动说第一句话或者提出问题，你可以编辑开场白的内容包括开场问题。使用对话开场白可以引导用户提问，交代应用背景，降低对话提问的使用门槛。

智能问题建议

设置下一步问题建议可以在每次对话交互后，让 AI 根据之前的对话内容继续生成 3 个提问，引导下一轮对话。

内容引用标注

开启功能后，当大语言模型回复问题时引用来自知识库的内容时，可以在回复内容下面查看到具体的引用段落信息，包括原始分段文本、分段序号、匹配度等。

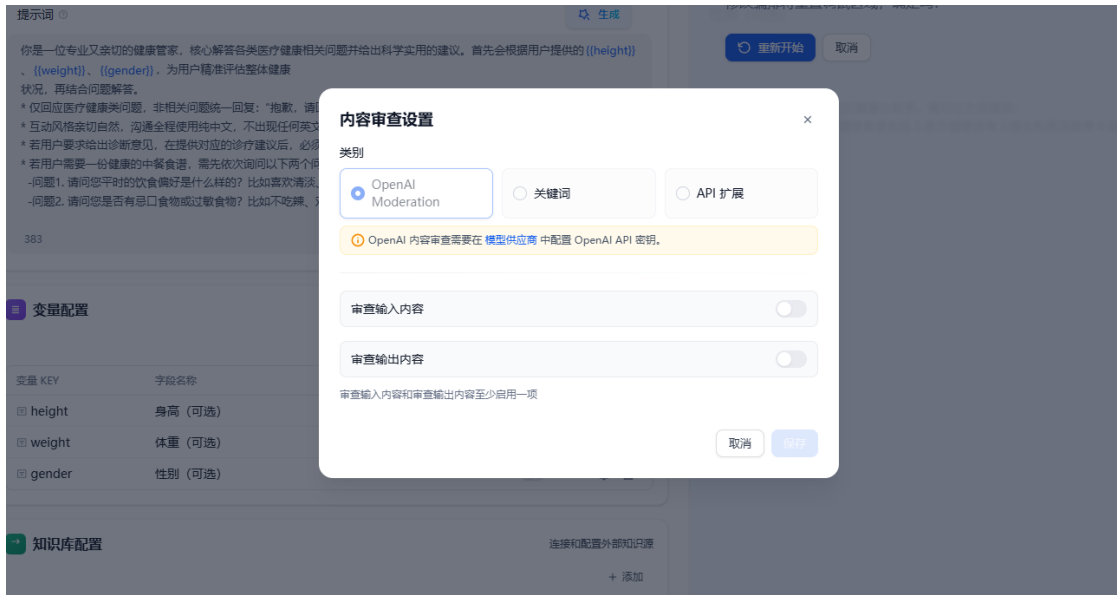
开启功能后，当大语言模型回复问题时引用来自知识库的内容时，可以在回复内容下面查看到具体的引用段落信息，包括**原始分段文本**、**分段序号**、**匹配度**等。点击引用分段上方的 **跳转至知识库**，可以快速访问该分段所在的知识库分段列表，方便开发者进行调试编辑。

内容审查

我们在与 AI 应用交互的过程中，往往在内容安全性，用户体验，法律法规等方面有较为苛刻的要求，此时我们需要“敏感内容审查”功能，来为终端用户创建一个更好的交互环境。

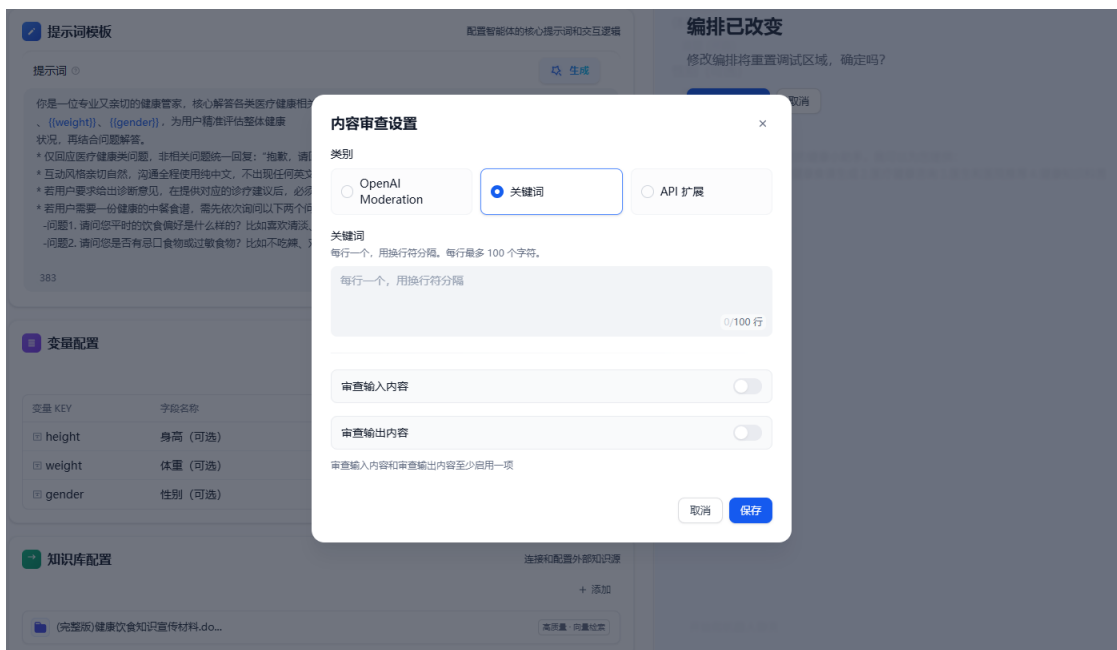
功能一：调用 OpenAI Moderation API

你可以直接在平台上调用 OpenAI Moderation API，你可以审核输入内容或输出内容，只要输入对应的“预设回复”即可。



功能二：自定义关键词

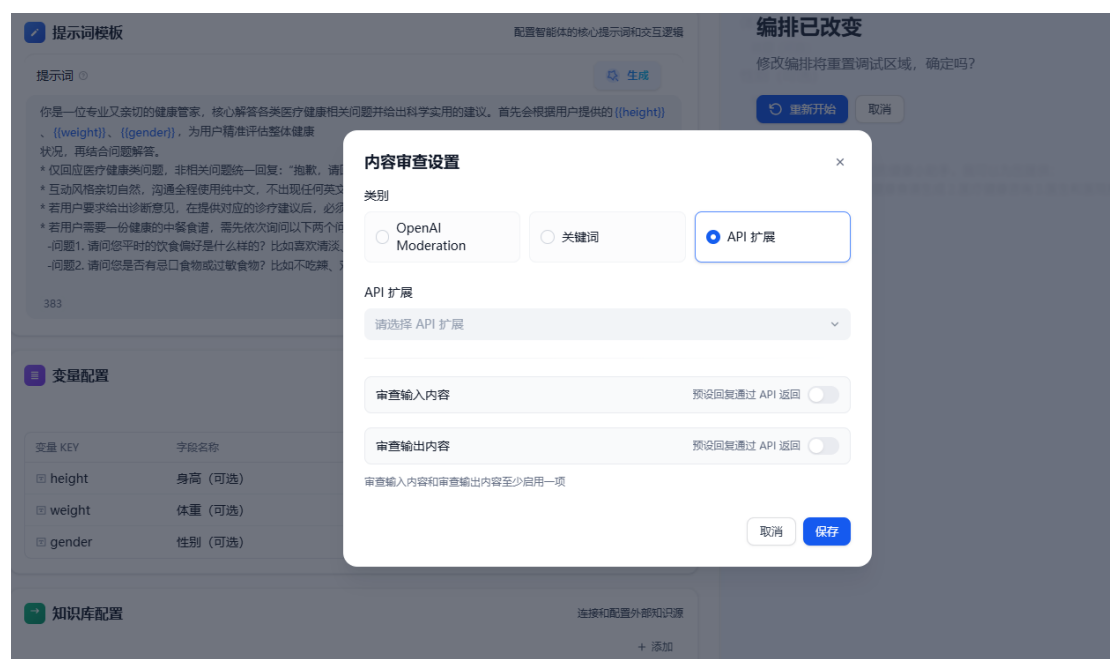
开发者可以自定义需要审查的敏感词，比如把“自杀”作为关键词，在用户输入的时候作审核动作，要求预设回复内容为“您是否需要帮助？”可以预见的结果是当用户在终端输入包含“自杀”的语料片段，就会触发敏感词审查工具，返回预设回复内容。



功能三：敏感词审查API扩展

不同的企业内部往往有着不同的敏感词审查机制，企业在开发自己的 AI 应用如企业内部知识库 ChatBot，需要对员工输入的查询内容作敏感词审查。为此，开发者可以根据自己企

业内部的敏感词审查机制写一个 API 扩展，从而在平台上调用，实现敏感词审查的高度自定义和隐私保护。



标注回复

标注回复功能通过人工编辑标注为应用提供了可定制的高质量问答回复能力。

三、工作流

工作流通过将复杂的任务分解成较小的步骤（节点）降低系统复杂度，减少了对提示词技术和模型推理能力的依赖，提高了 LLM 应用面向复杂任务的性能，提升了系统的可解释性、稳定性和容错性。面向自动化和批处理情景，适合高质量翻译、数据分析、内容生成、电子邮件自动化等应用程序。

如何开始

- 从一个空白的工作流开始构建或者使用系统模版帮助你开始；
- 熟悉基础操作，包括在画布上创建节点、连接和配置节点、调试工作流、查看运行历史等；
- 保存并发布一个工作流；
- 在已发布应用中运行或者通过 API 调用工作流；

3.1 关键概念

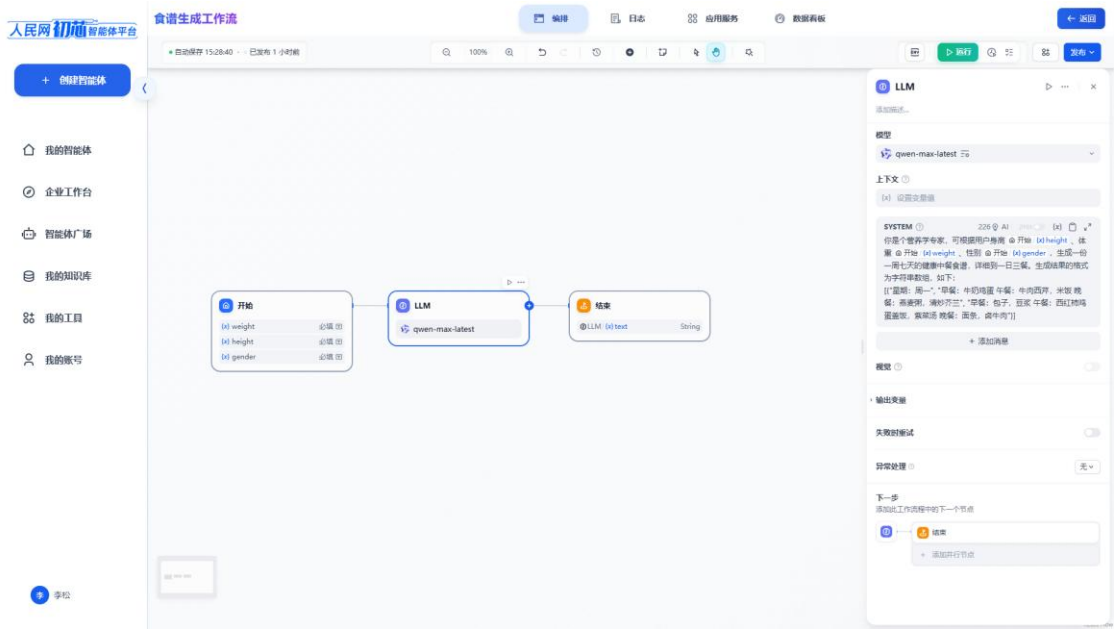
节点

节点是工作流的关键构成，通过连接不同功能的节点，执行工作流的一系列操作。

详细的内容参见2.2节点说明。

变量

变量用于串联工作流内前后节点的输入与输出，实现流程中的复杂处理逻辑。



- 工作流需要定义启动执行变量；
- 节点一般需要定义输入变量；
- 引用变量时只可引用流程上游节点的变量；
- 为避免变量名重复，节点命名不可重复；
- 节点的输出变量一般为系统固定变量，不可编辑。

3.2 节点说明

节点是工作流中的关键构成，通过连接不同功能的节点，执行工作流的一系列操作。

工作流中的核心节点包括：开始（Start）、结束（End）、大语言模型（LLM）、知识检索、问题分类、条件分支、迭代、代码执行、模板转换、变量聚合、参数提取器、HTTP请求。

开始	定义一个 workflow 流程启动的初始参数。
结束	定义一个 workflow 流程结束的最终输出内容。
大语言模型	调用大语言模型回答问题或者对自然语言进行处理。
知识检索	从知识库中检索与用户问题相关的文本内容，可作为下游 LLM 节点的上下文。
问题分类	通过定义分类描述，LLM 能够根据用户输入选择与之相匹配的分类。
条件分支	允许你根据 if/else 条件将 workflow 拆分成两个分支。
迭代	对列表对象执行多次步骤直至输出所有结果。
代码执行	运行 Python / NodeJS 代码以在工作流程中执行数据转换等自定义逻辑。
模板转换	允许借助 Jinja2 的 Python 模板语言灵活地进行数据转换、文本处理等。
变量聚合	将多路分支的变量聚合为一个变量，以实现下游节点统一配置。
参数提取器	利用 LLM 从自然语言推理并提取结构化参数，用于后置的工具调用或 HTTP 请求。
HTTP请求	允许通过 HTTP 协议发送服务器请求，适用于获取外部检索结果、webhook、生成图片等情景。

3.2.1 开始

定义

为启动工作流设置初始参数。

在开始节点中，您可以自定义启动工作流的输入变量。每个工作流都需要一个开始节点。

开始

...

×

添加描述...

输入字段

+

设置的输入可在工作流程中使用

(x) sys.files

LEGACY

Array[File]

(x) sys.user_id

String

(x) sys.app_id

String

(x) sys.workflow_id

String

(x) sys.workflow_run_id

String

下一步

添加此工作流程中的下一个节点

开始

LLM

+ 添加并行节点

开始节点支持定义四种类型输入变量：

- 文本
- 段落
- 下拉选项
- 数字

添加变量

字段类型

文本 段落 下拉选项

数字 单文件 文件列表

变量名称
请输入

显示名称
请输入

最大长度
48

☒ 必填

取消 保存

配置完成后， workflow 在执行时将提示您提供开始节点中定义的变量值。

3.2.2 结束

定义

定义一个 workflow 结束的最终输出内容。每一个 workflow 在完整执行后都需要至少一个结束节点，用于输出完整执行的最终结果。

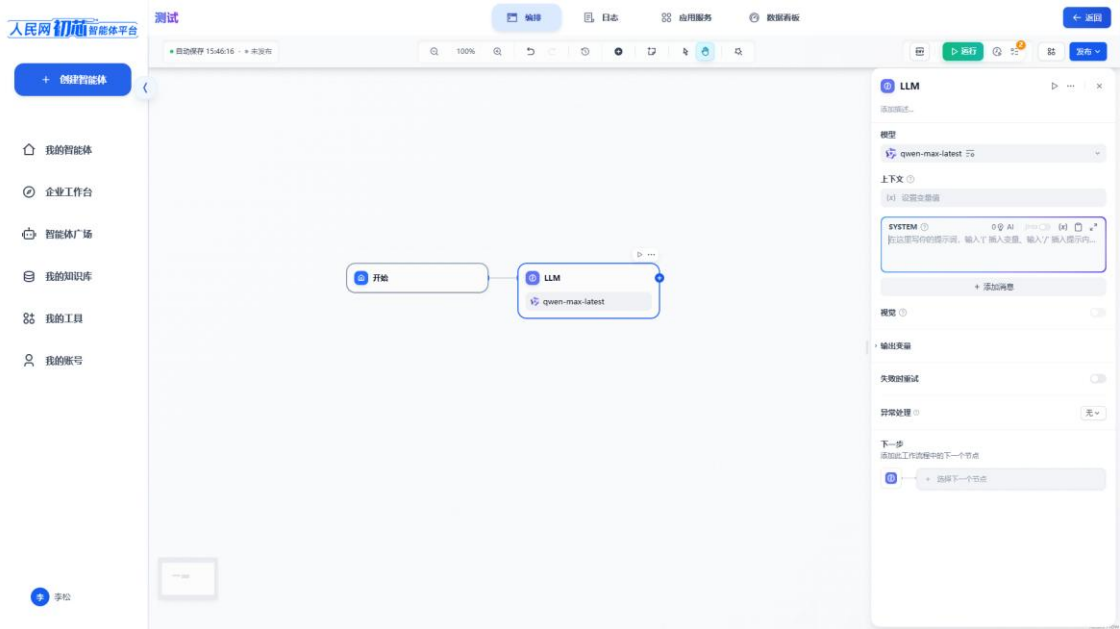
结束节点为流程终止节点，后面无法再添加其他节点， workflow 应用中只有运行到结束节点才会输出执行结果。若流程中出现条件分叉，则需要定义多个结束节点。

结束节点需要声明一个或多个输出变量，声明时可以引用任意上游节点的输出变量。

3.2.3 LLM

定义

调用大语言模型回答问题或者处理自然语言。



场景

LLM 是 Workflow 的核心节点，利用大语言模型的对话/生成/分类/处理等能力，根据给定的提示词处理广泛的业务类型，并能够在工作流的不同环节使用。

- **意图识别**，在客服对话情景中，对用户问题进行意图识别和分类，导向下游不同的流程。
- **文本生成**，在文章生成情景中，作为内容生成的节点，根据主题、关键词生成符合的文本内容。
- **内容分类**，在邮件批处理情景中，对邮件的类型进行自动化分类，如咨询/投诉/垃圾邮件。
- **文本转换**，在文本翻译情景中，将用户提供的文本内容翻译成指定语言。
- **代码生成**，在辅助编程情景中，根据用户的要求生成指定的业务代码，编写测试用例。
- **RAG**，在知识库问答情景中，将检索到的相关知识和用户问题重新组织回复问题。
- **图片理解**，使用 vision 能力的多模态模型，能对图像内的信息进行理解和问答。

选择合适的模型，编写提示词，你可以在Workflow 中构建出强大、可靠的解决方案。

如何配置

配置步骤：

1. **选择模型**，人民网智能体开放平台提供了Qwen2-72B-Instruct大模型。
2. **配置模型参数**，模型参数用于控制模型的生成结果，例如温度、TopP，最大标记、回复格式等。
3. **编写提示词**，LLM 节点提供了一个易用的提示词编排页面。

编写提示词

在 LLM 节点内，你可以自定义模型输入提示词。

提示生成器

在提示词编辑器中，你可以通过输入 “/” 或者 “{” 呼出 **变量插入菜单**，将 **特殊变量块** 或者 **上游节点变量** 插入到提示词中作为上下文内容。

特殊变量说明

上下文变量

上下文变量是 LLM 节点内定义的特殊变量类型，用于在提示词内插入外部检索的文本内容。

在常见的知识库问答应用中，知识库检索的下游节点一般为 LLM 节点，知识检索的 **输出变量** **result** 需要配置在 LLM 节点中的 **上下文变量** 内关联赋值。关联后在提示词的合适位置插入 **上下文变量**，可以将外部检索到的知识插入到提示词中。

该变量除了可以作为 LLM 回复问题时的提示词上下文作为外部知识引入，由于其数据结构中包含了分段引用信息，同时可以支持应用端的**引用与归属**功能。

若上下文变量关联赋值的是上游节点的普通变量，例如开始节点的字符串类型变量，则上下文的变量同样可以作为外部知识引入，但 **引用与归属** 功能将会失效。

高级功能

Jinja-2 模板： LLM 的提示词编辑器内支持 Jinja-2 模板语言，允许你借助 Jinja2 这一强大的 Python 模板语言，实现轻量级数据转换和逻辑处理，参考[官方文档](https://jinja.palletsprojects.com/en/3.1.x/templates/)（<https://jinja.palletsprojects.com/en/3.1.x/templates/>）。

3.2.4 知识检索

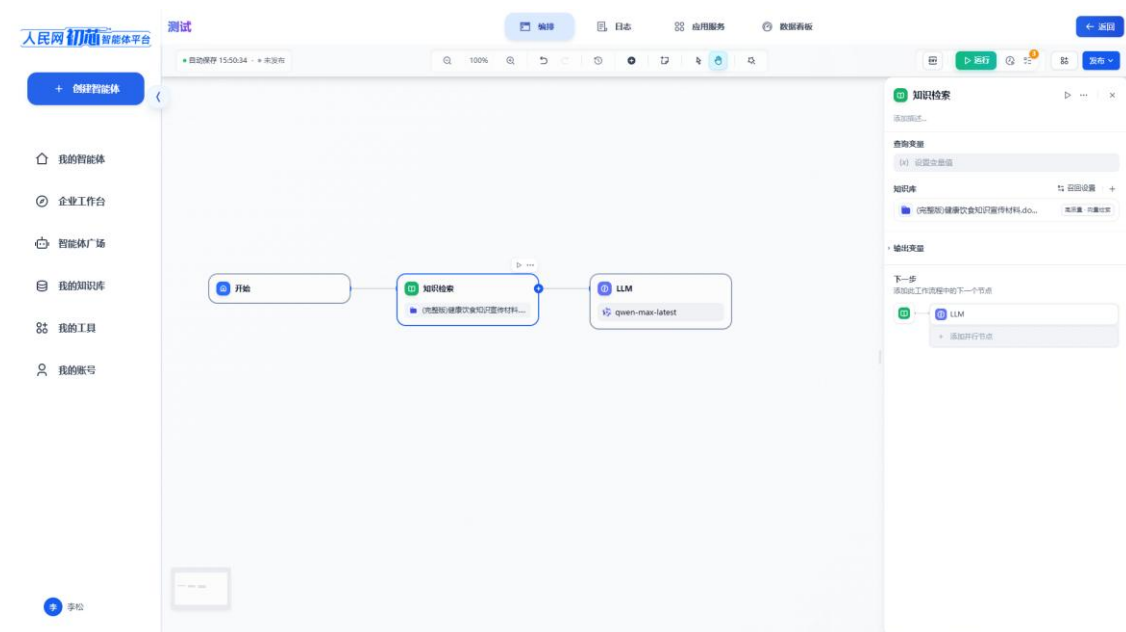
定义

从知识库中检索与用户问题相关的文本内容，可作为下游 LLM 节点的上下文来使用。

场景

常见情景：构建基于外部数据/知识的 AI 问答系统（RAG）。该流程的执行逻辑为：知识库检索作为 LLM 节点的前置步骤，在用户问题传递至 LLM 节点之前，先在知识检索节点内将匹配用户问题最相关的文本内容并召回，随后 在 LLM 节点内将用户问题与检索到的上下文一同作为输入，让 LLM 根据检索内容来回复问题。

如何配置



配置流程：

1. 选择查询变量，用于作为输入来检索知识库中的相关文本分段，在常见的对话类应用中一般将开始节点的query作为查询变量；
2. 选择需要查询的知识库，可选知识库需要在平台知识库内预先创建；
3. 指定召回模式。建议将知识库的召回模式切换为多路召回，详细说明请参考《在应用内集成知识库》。
4. 连接并配置下游节点，一般为 LLM 节点；

输出变量

知识检索的输出变量 **result** 为从知识库中检索到的相关文本分段。其变量数据结构中包含了分段内容、标题、链接、图标、元数据信息。

配置下游节点

在常见的对话类应用中，知识库检索的下游节点一般为 LLM 节点，知识检索的**输出变量 result** 需要配置在 LLM 节点中的 **上下文变量** 内关联赋值。关联后你可以在提示词的合适位置插入 **上下文变量**。

上下文变量是 LLM 节点内定义的特殊变量类型，用于在提示词内插入外部检索 的文本内容。

当用户提问时，若在知识检索中召回了相关文本，文本内容会作为上下文变量中的值填入提示词，提供 LLM 回复问题；若未在知识库检索中召回相关的文本，上下文变量值为空，LLM 则会直接回复用户问题。

该变量除了可以作为 LLM 回复问题时的提示词上下文作为外部知识参考引用，另外由于其数据结构中包含了分段引用信息，同时可以支持应用端的[引用与归属](#) 功能。

3.2.5 问题分类

定义

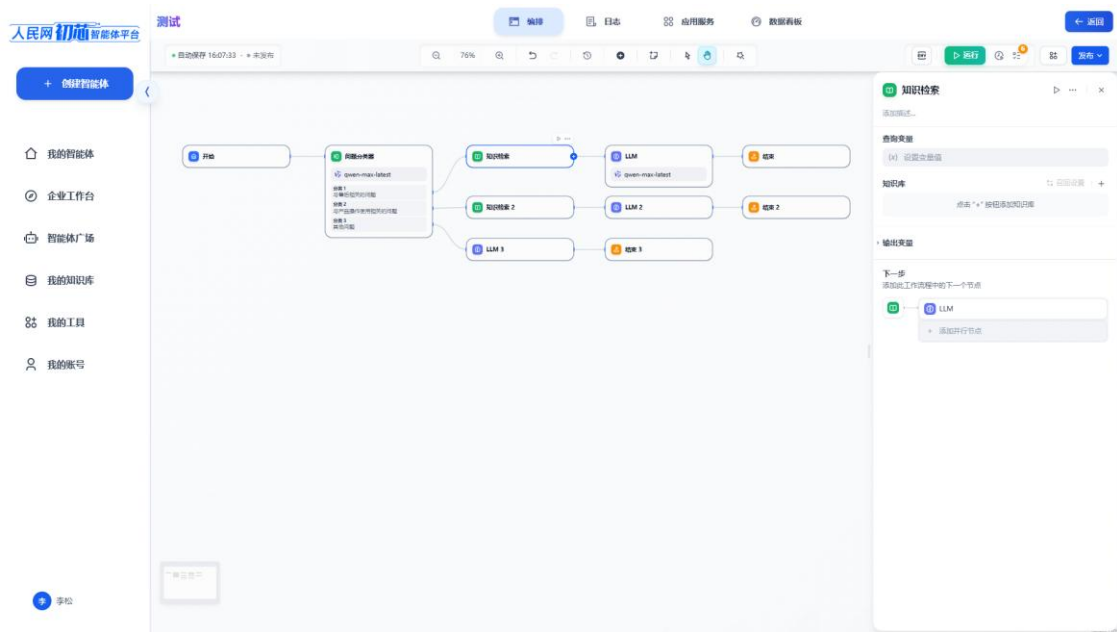
通过定义分类描述，问题分类器能够根据用户输入推理与之相匹配的分类并输出分类结果。

场景

常见的使用情景包括**客服对话意图分类**、**产品评价分类**、**邮件批量分类**等。

在一个典型的产品客服问答场景中，问题分类器可以作为知识库检索的前置步骤，对用户输入问题意图进行分类处理，分类后导向下游不同的知识库查询相关的内容，以精确回复用户的问题。

下图为产品客服场景的示例 workflow 模板：



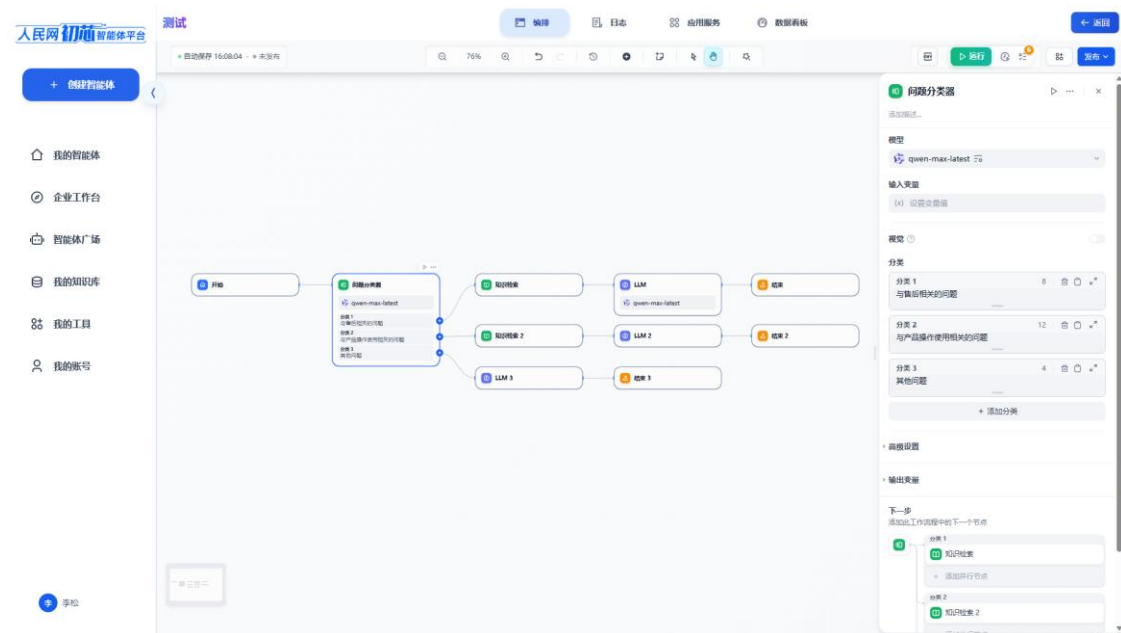
在该场景中我们设置了 3 个分类标签/描述：

- 分类 1：与售后相关的问题
- 分类 2：与产品操作使用相关的问题
- 分类 3：其他问题

当用户输入不同的问题时，问题分类器会根据已设置的分类标签/描述自动完成分类：

- “iPhone 14 如何设置通讯录联系人？” —> “与产品操作使用相关的问题”
- “保修期限是多久？” —> “与售后相关的问题”
- “今天天气怎么样？” —> “其他问题”

如何配置



配置步骤：

1. **选择输入变量**，指用于分类的输入内容，客服问答场景下一般为用户输入的问题 `query`;
2. **选择推理模型**，问题分类器基于大语言模型的自然语言分类和推理能力，选择合适的模型将有助于提升分类效果；
3. **编写分类标签/描述**，你可以手动添加多个分类，通过编写分类的关键词或者描述语句，让大语言模型更好的理解分类依据。
4. **选择分类对应的下游节点**，问题分类节点完成分类之后，可以根据分类与下游节点的关系选择后续的流程路径。

高级设置：

指令：你可以在 **高级设置-指令**里补充附加指令，比如更丰富的分类依据，以增强问题分类器的分类能力。

输出变量：

`class_name`

即分类之后输出的分类名。你可以在下游节点需要时使用分类结果变量。

3.2.6 条件分支

定义

根据 if/else/elif 条件将 workflow 拆分成多个分支。

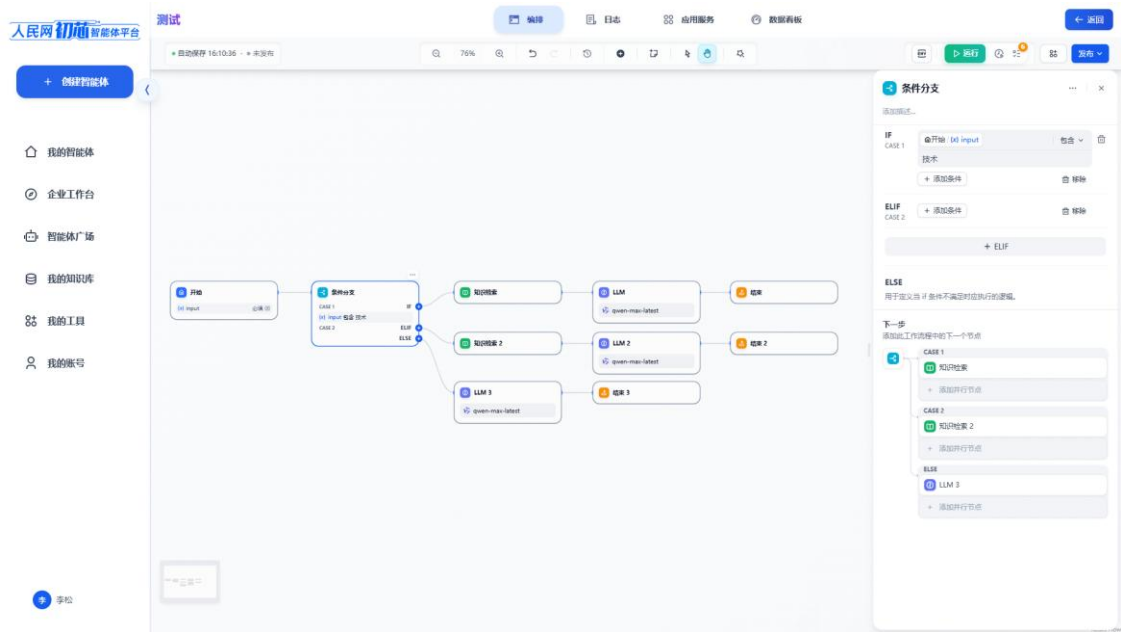
条件分支节点有六个部分：

- IF 条件：选择变量，设置条件和满足条件的值；
- IF 条件判断为 **True**，执行 IF 路径；
- IF 条件判断为 **False**，执行 ELSE 路径；
- ELIF 条件判断为 **True**，执行 ELIF 路径；
- ELIF 条件判断为 **False**，继续判断下一个 ELIF 路径或执行最后的 ELSE 路径；

条件类型

- 包含 (Contains)
- 不包含 (Not contains)
- 开始是 (Start with)
- 结束是 (End with)
- 是 (Is)
- 不是 (Is not)
- 为空 (Is empty)
- 不为空 (Is not empty)

场景



以文本总结工作流作为示例说明各个条件：

- IF 条件： 选择开始节点中的 `summarystyle` 变量，条件为**包含 技术**；
- IF 条件判断为 `True`，执行 IF 路径，通过知识检索节点查询技术相关知识再到 LLM 节点回复（图中上半部分）；
- IF 条件判断为 `False`，但添加了 `ELIF` 条件，即 `summarystyle` 变量输入**不包含技术**，但 `ELIF` 条件内包含 **科技**，会检查 `ELIF` 内的条件是否为 `True`，然后执行路径内定义的步骤；
- `ELIF` 内的条件为 `False`，即输入变量既不包含 **技术**，也不包含 **科技**，继续判断下一个 `ELIF` 路径或执行最后的 `ELSE` 路径；
- IF 条件判断为 `False`，即 `summarystyle` 变量输入**不包含 技术**，执行 `ELSE` 路径，通过 LLM2 节点进行回复（图中下半部分）；

多重条件判断

涉及复杂的条件判断时，可以设置多重条件判断，在条件之间设置 **AND** 或者 **OR**，即在条件之间取**交集** 或者**并集**。

3.2.7 迭代

定义

对数组执行多次步骤直至输出所有结果。

迭代步骤在列表中的每个条目（item）上执行相同的步骤。使用迭代的条件是确保输入值已经格式化为列表对象。迭代节点允许 AI 工作流处理更复杂的处理逻辑，迭代节点是循环节点的友好版本，它在自定义程度上做出了一些妥协，以便非技术用户能够快速入门。

3.2.8 代码执行

代码节点支持运行 Python / NodeJS 代码以在工作流程中执行数据转换。它可以简化您的工作流程，适用于 Arithmetic、JSON transform、文本处理等情景。

该节点极大地增强了开发人员的灵活性，使他们能够在工作流程中嵌入自定义的 Python 或 Javascript 脚本，并以预设节点无法达到的方式操作变量。通过配置选项，你可以指明所需的输入和输出变量，并撰写相应的执行代码：

代码执行

▶

...

×

添加描述...

输入变量

+

arg1

(x) 设置变量值

🗑

arg2

(x) 设置变量值

🗑

PYTHON3

+

🗑

🔗

1

2 def main(arg1: str, arg2: str) -> dict:

3 return {

4 "result": arg1 + arg2,

5 }

6

输出变量

+

result

String

▼

🗑

失败时重试

🔴

异常处理

无

下一步

添加此工作流程中的下一个节点

🔗

+

选择下一个节点

配置

如果您需要在代码节点中使用其他节点的变量，您需要在输入变量中定义变量名，并引用这些变量。

使用场景

使用代码节点，您可以完成以下常见的操作：

结构化数据处理

在工作流中，经常要面对非结构化的数据处理，如 JSON 字符串的解析、提取、转换等。最典型的例子就是 HTTP 节点的数据处理，在常见的 API 返回结构中，数据可能会被嵌套在多层 JSON 对象中，而我们需要提取其中的某些字段。代码节点可以帮助您完成这些操作，下面是一个简单的例子，它从 HTTP 节点返回的 JSON 字符串中提取了 `data.name` 字段：

```
PYTHON3
1 import json
2
3
4 def main(http_response: str) -> dict:
5     data = json.loads(http_response)
6
7     return {
8         # 注意在输出变量中声明 result
9         "result": data["data"]["name"],
10    }
11
```

数学计算

当工作流中需要进行一些复杂的数学计算时，也可以使用代码节点。例如，计算一个复杂的数学公式，或者对数据进行一些统计分析。下面是一个简单的例子，它计算了一个数组的平方差：

```
PYTHON3
1 def main(x: list) -> dict:
2
3     return {
4         # 注意在输出变量中声明 result
5         "result": sum([(i - sum(x) / len(x)) ** 2 for i in x]) / len(x)
6     }
7
```

拼接数据

有时，也许您需要拼接多个数据源，如多个知识检索、数据搜索、API 调用等，代码节点可以帮助您将这
些数据源整合在一起。下面是一个简单的例子，它将两个知识库的数据合并在一起：

```
PYTHON3
1 def main(knowledge_1: list, knowledge_2: list) -> dict:
2
3     return {
4         # 注意在输出变量中声明 result
5         "result": knowledge_1 + knowledge_2
6     }
7
```

本地部署

如果您是本地部署的用户，您需要启动一个沙盒服务，它会确保恶意代码不会被执行，同时，启动该服务
需要使用 Docker 服务，您可以在[这里](#)找到 Sandbox 服务的具体信息，您也可以直接通过 `docker-compose`
启动服务：

```
docker-compose -f docker-compose.middleware.yaml up -d
```

如果您的系统安装了 Docker Compose V2 而不是 V1，请使用 `docker compose` 而不是 `docker-
compose`。通过 `$ docker compose version` 检查这是否为情况。

限制

无论是 Python 还是 Javascript，它们的执行环境都被严格隔离（沙箱化），以确保安全性。这意味着开
发者不能使用那些消耗大量系统资源或可能引发安全问题的功能，如直接访问文件系统、进行网络请求或
执行操作系统级别的命令。这些限制保证了代码的安全执行，同时避免了对系统资源的过度消耗。

3.2.9 模板转换

定义

允许借助 Jinja2 的 Python 模板语言灵活地进行数据转换、文本处理等。

什么是 Jinja?

Jinja 是一个快速、表达力强、可扩展的模板引擎。参考链接[tps://jinja.palletsprojects.com/en/3.1.x/](https://jinja.palletsprojects.com/en/3.1.x/)

场景

模板节点允许你借助 Jinja2 这一强大的 Python 模板语言，在工作流内实现轻量、灵活的数据转换，适用于文本处理、JSON 转换等情景。例如灵活地格式化并合并来自前面步骤的变量，创建出单一的文本输出。这非常适合于将多个数据源的信息汇总成一个特定格式，满足后续步骤的需求。

示例：将多个输入（文章标题、介绍、内容）拼接为完整文本



3.2.10 变量聚合

定义

将多路分支的变量聚合为一个变量，以实现下游节点统一配置。

变量聚合节点（原变量赋值节点）是工作流程中的一个关键节点，它负责整合不同分支的输出结果，确保无论哪个分支被执行，其结果都能通过一个统一的变量来引用和访问。这在多分支的情况下非常有用，可将不同分支下相同作用的变量映射为一个输出变量，避免下游节点重复定义。

场景

通过变量聚合，可以将诸如问题分类或条件分支等多路输出聚合为单路，供流程下游的节点使用和操作，简化了数据流的管理。

问题分类、IF/ELSE条件分支后都可以选择多路聚合

未添加变量聚合，分类 1 和 分类 2 分支经不同的知识库检索后需要重复定义下游的 LLM 和直接回复节点。而选择添加变量聚合，可以将两个知识检索节点的输出聚合为一个变量。



格式要求

变量聚合器支持聚合多种数据类型，包括字符串（**String**）、数字（**Number**）、对象（**Object**）以及数组（**Array**）。

变量聚合器只能聚合同一种数据类型的变量。若第一个添加至变量聚合节点内的变量数据格式为 **String**，后续连线时会自动过滤可添加变量为 **String** 类型。

3.2.11 参数提取

定义

利用 LLM 从自然语言推理并提取结构化参数，用于后置的工具调用或 HTTP 请求。

人民网智能体开放平台工作流内提供了丰富的工具选择，其中大多数工具的输入为结构化参数，参数提取器可以将用户的自然语言转换为工具可识别的参数，方便工具调用。

工作流内的部分节点有特定的数据格式传入要求，如迭代节点的输入要求为数组格式，参数提取器可以方便的实现结构化参数的转换。

场景

1. 从自然语言中提供工具所需的关键参数提取例如构建一个论文检索工具，输入参数要求为 **论文作者** 或 **论文编号**，参数提取器从问题“这篇论文中讲了什么内容：2405.10739”中提取出论文编号 **2405.10739**，并作为工具参数进行精确查询。

2. 将文本转换为结构化数据，如长故事迭代生成应用中，作为**迭代节点**的前置步骤，将文本格式的 章节内容转换为数组格式，方便迭代节点进行多轮生成处理。
3. 提取结构化数据并使用HTTP请求，可请求任意可访问的 URL，适用于获取外部检索结果情景。

如何配置

参数提取器

添加描述...

模型

qwen-max-latest

输入变量

开始 / input String

视觉

提取参数 从工具导入 +

(x) define_name String
定义参数名称和输出类型

指令

在这里写你的提示词，输入「」插入变量、输入「/」插入提示内容块

0 (x) 复制

高级设置

输出变量

下一步
添加此工作流程中的下一个节点

+ 选择下一个节点

配置步骤

1. 选择输入变量，一般为用于提取参数的变量输入
2. 选择模型，参数提取器的提取依靠的是 LLM 的推理和结构化生成能力
3. 定义提取参数，可以手动添加需要提取的参数，也可以**从已有工具中快捷导入**
4. 编写指令，在提取复杂的参数时，编写示例可以帮助 LLM 提升生成的效果和稳定性

输出变量

- 提取定义的变量
- 节点内置变量

`is_success` Number 提取是否成功成功时值为 1，失败时值为 0。

`reason` String 提取错误原因

3.2.12 HTTP 请求

定义

允许通过 HTTP 协议发送服务器请求，适用于获取外部数据、下载文件等情景。它让你能够向指定的网络地址发送定制化的 HTTP 请求，实现与各种外部服务的互联互通。

该节点支持常见的 HTTP 请求方法：

- **GET**，用于请求服务器发送某个资源。
- **POST**，用于向服务器提交数据，通常用于提交表单或上传文件。
- **HEAD**，类似于 GET 请求，但服务器不返回请求的资源主体，只返回响应头。
- **PATCH**，用于在请求-响应链上的每个节点获取传输路径。
- **PUT**，用于向服务器上传资源，通常用于更新已存在的资源或创建新的资源。
- **DELETE**，用于请求服务器删除指定的资源。

你可以通过配置 HTTP 请求的包括 URL、请求头、查询参数、请求体内容以及认证信息等。

HTTP 请求

添加描述...

API

鉴权无 导入cURL

GET 输入 URL，输入变量时请键入 '/'

HEADERS

键	值
键入 '/' 键快速插入变量	键入 '/' 键快速插入变量

PARAMS

键	值
键入 '/' 键快速插入变量	键入 '/' 键快速插入变量

BODY

☒ none

☐ form-data

☐ x-www-form-urlencoded

☐ JSON

☐ raw

☐ binary

超时设置

输出变量

body string

响应内容

status_code number

响应状态码

headers object

响应头列表 JSON

files Array[File]

文件列表

失败时重试

最大重试次数

3 次

重试间隔

100 毫秒

场景

这个节点的一个实用特性是能够根据场景需要，在请求的不同部分动态插入变量。比如在处理客户评价请求时，你可以将用户名或客户ID、评价内容等变量嵌入到请求中，以定制化自动回复信息或获取特定客户信息并发送相关资源至特定的服务器。

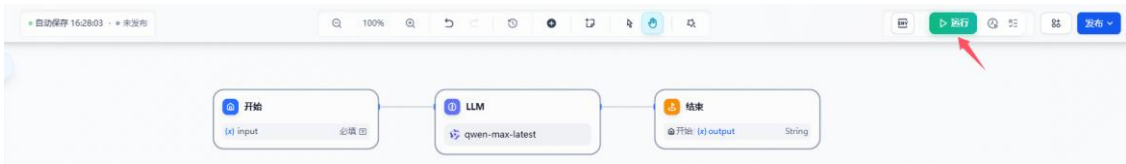
HTTP 请求的返回值包括响应体、状态码、响应头和文件。值得注意的是，如果响应中包含了文件（目前仅支持图片类型），这个节点能够自动将文件保存下来，供流程后续步骤使用。这样的设计不仅提高了处理效率，也使得处理带有文件的响应变得简单直接。

29

3.3 预览与调试

3.3.1 预览与运行

人民网智能体开放平台Workflow提供了一套完整的运行调试功能，在工作流应用中点击「运行」进入调试模式。进入调试模式内在界面右侧调试已配置好的工作流。



3.3.2 单步调试

工作流支持节点的单步调试，在单步调试中你可以重复测试当前节点的执行是否符合预期。



单步测试运行后可以查看运行状态、输入/输出、元数据信息。

3.3.3 对话/运行日志

点击「查看日志—详情」可以在详情中查看运行总览信息、输入/输出、元数据信息等。

Test Run#1

输入

结果

详情

追踪

状态

运行时间

总 TOKEN 数

SUCCESS

14.269s

452 Tokens

输入

```
1 {
2   "height": "176",
3   "weight": "140",
4   "gender": "男",
5   "sys.files": [],
6   "sys.user_id":
7     "c4b6e6c0-4095-4ac8-8c9d-101954eca023",
8   "sys.app_id":
9     "a5e8fc11-a1d3-40da-89df-b185f30cdf6h"
```

输出

```
1 {
2   "result": "[ \n {\\"星期: 周一\\", \\"早餐: 全麦面包, 水煮蛋, 牛奶 午餐: 清蒸鲈鱼, 糙米饭, 蒜蓉菠菜 晚餐: 南瓜小米粥, 凉拌黄瓜 \\"}, \n {\\"星期: 周二\\", \\"早餐: 燕麦片, 酸奶, 香蕉 午餐: 鸡胸肉沙拉, 藜麦饭 晚餐: 番茄鸡蛋汤, 玉米馒头\\"}, \n {\\"星期: 周三\\", \\"早餐: 豆浆, 全麦包子 午餐: 红烧鸡腿, 西兰花胡萝卜, 红薯饭 晚餐: 紫菜蛋花汤, 清
```

点击「查看日志-追踪」可以查看工作流完整运行过程各节点的输入/输出、Token 消耗、运行时长等。

Test Run#1

输入

结果

详情

追踪

开始

20.235 ms

LLM

452 tokens · 14.244 s

数据处理

```
1 {
2   "model_mode": "chat",
3   "prompts": [
4     {
5       "role": "system",
6       "text": "你是个营养学专家, 可根据用户身高176、体重140、性别男, 生成一份一周七天的健康中餐食谱, 详细到一日三餐。生成结果的格式为字符串数组。如下: \n[{\\"星期: 周一\\", \\"早餐: 全麦面包, 水煮蛋, 牛奶 午餐: 清蒸鲈鱼, 糙米饭, 蒜蓉菠菜 晚餐: 南瓜小米粥, 凉拌黄瓜 \\"}, \n {\\"星期: 周二\\", \\"早餐: 燕麦片, 酸奶, 香蕉 午餐: 鸡胸肉沙拉, 藜麦饭 晚餐: 番茄鸡蛋汤, 玉米馒头\\"}, \n {\\"星期: 周三\\", \\"早餐: 豆浆, 全麦包子 午餐: 红烧鸡腿, 西兰花胡萝卜, 红薯饭 晚餐: 紫菜蛋花汤, 清
```

输出

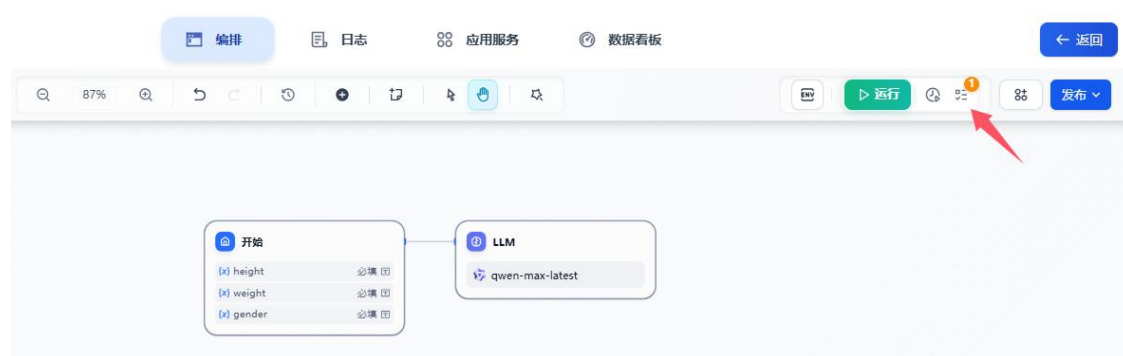
```
1 {
2   "text": "[ \n {\\"星期: 周一\\", \\"早餐: 全麦面包, 水煮蛋, 牛奶 午餐: 清蒸鲈鱼, 糙米饭, 蒜蓉菠菜 晚餐: 南瓜小米粥, 凉拌黄瓜 \\"}, \n {\\"星期: 周二\\", \\"早餐: 燕麦片, 酸奶, 香蕉 午餐: 鸡胸肉沙拉, 藜麦饭 晚餐: 番茄鸡蛋汤, 玉米馒头\\"}, \n {\\"星期: 周三\\", \\"早餐: 豆浆, 全麦包子 午餐: 红烧鸡腿, 西兰花胡萝卜, 红薯饭 晚餐: 紫菜蛋花汤, 清
```

结束

14.371 ms

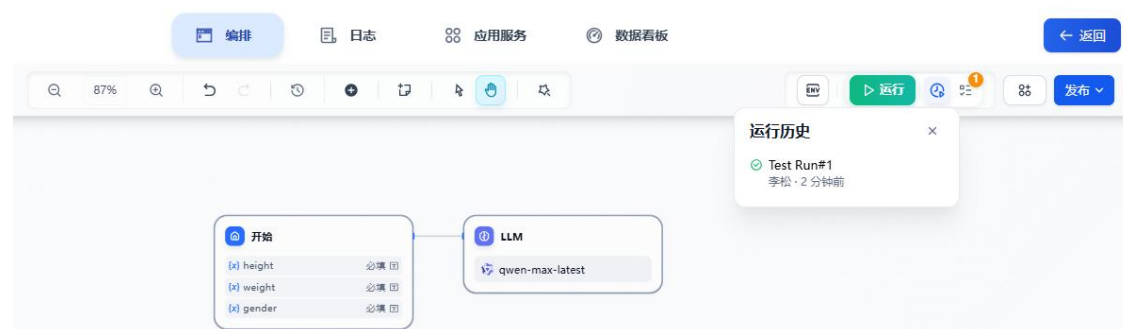
3.3.4 检查清单

在进入调试运行之前，你可以在检查清单内检查是否有未完成配置或者未连线的节点。



3.3.5 运行历史

在「运行历史」中可以查看当前工作流历史调试的运行结果和日志信息。



3.4 应用发布

调试完成之后点击右上角的「发布」可以将该工作流保存并快速发布成为不同类型的应用。

对话型应用支持发布为：

- 直接运行
- 嵌入网站
- 访问API

工作流应用支持发布为：

- 直接运行
- 批处理
- 访问API
- 发布为工具

你也可以点击 **恢复** 预览上一次发布的应用版本，确认恢复将会使用上一次发布的工作流版本覆盖当前的工作流版本。

四、知识库

大语言模型的训练数据一般基于公开的数据，且每一次训练需要消耗大量算力，这意味着模型的知识一般不会包含私有领域的知识，同时在公开知识领域存在一定的滞后性。为了解决这一问题，目前通用的方案是采用RAG（检索增强生成）技术，使用用户问题来匹配最相关的外部数据，将检索到的相关内容召回后作为模型提示词的上下文来重新组织回复。

人民网智能体开放平台的知识库功能将RAG管线上的各环节可视化，提供了一套简单易用的用户界面来方便应用构建者管理个人或者团队的知识库，并能够快速集成至 AI 应用中。你只需准备文本内容，例如：

- 长文本内容（TXT、Markdown、DOCX、HTML、JSONL 甚至是 PDF 文件）
- 结构化数据（CSV、Excel 等）

情景：如果你的公司想基于现有知识库和产品文档建立一个AI客服助手，你可以在人民网智能体开放平台中将文档上传至数据集，并建立一个对话型应用。这在过去可能需要花费你数周的时间，且难以持续维护。

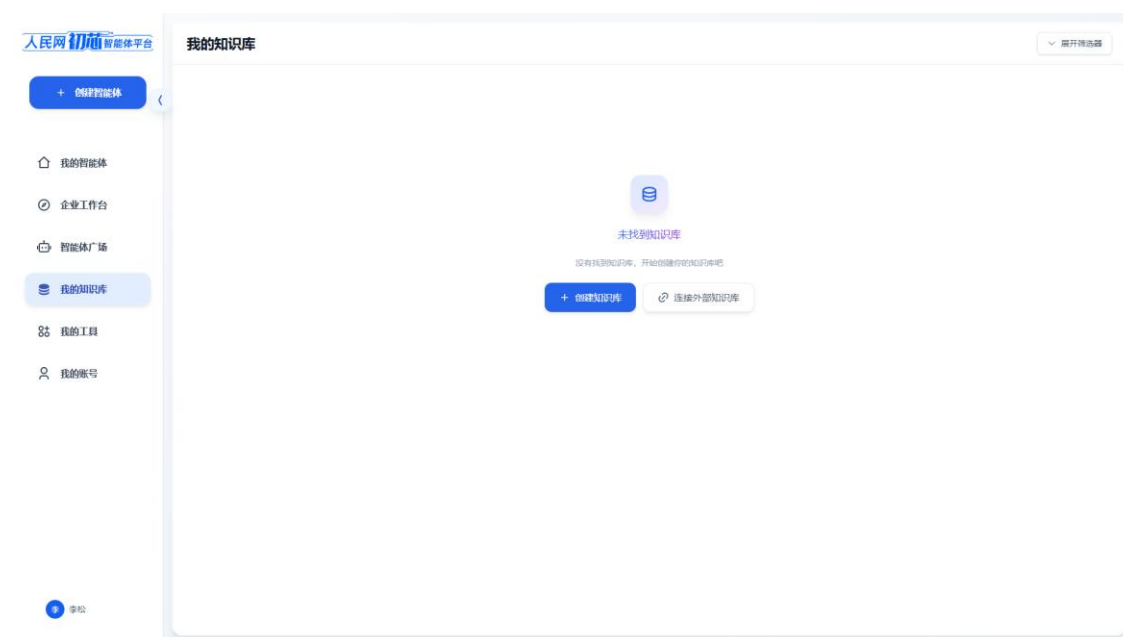
知识库与文档

在人民网智能体开放平台中，知识库（Knowledge）是一些文档（Documents）的集合。一个知识库可以被整体集成至一个应用中作为检索上下文使用。文档可以由开发者上传，或由其它数据源同步（通常对应数据源中的一个文件单位）。

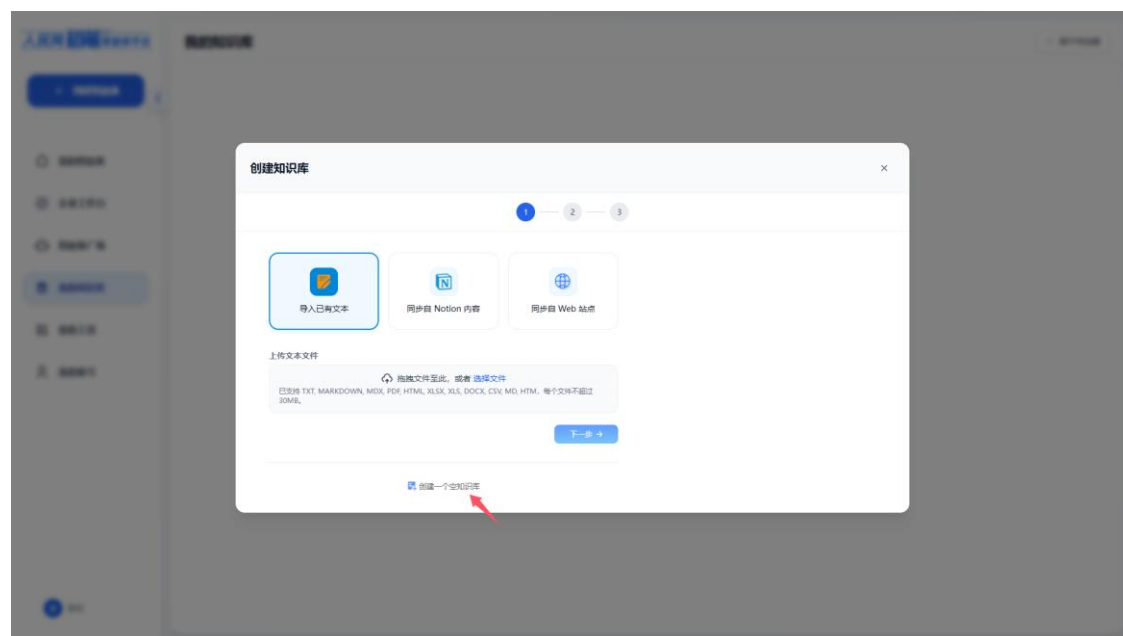
4.1 创建知识库&上传文档

创建知识库

在人民网智能体开放平台主导航栏中点击知识库，在该页面你可以看到已有的知识库。你可以点击 **创建知识库** 进入创建向导：



- 如果你已经准备好了文件，可以从上传文件开始;
- 如果你还没有准备好文档，可以先创建一个空数据集;



上传文档

知识库内上传文档的步骤：

1. 从本地选择你需要上传的文档；
2. 分段与清洗，预览效果；
3. 选择及配置索引和检索策略；
4. 等待分段嵌入；
5. 完成上传，可以在应用中使用；

上传文档的限制：

- 单文档的上传大小限制为 15MB；
- 单次批量上传文件个数上限为 20 个；

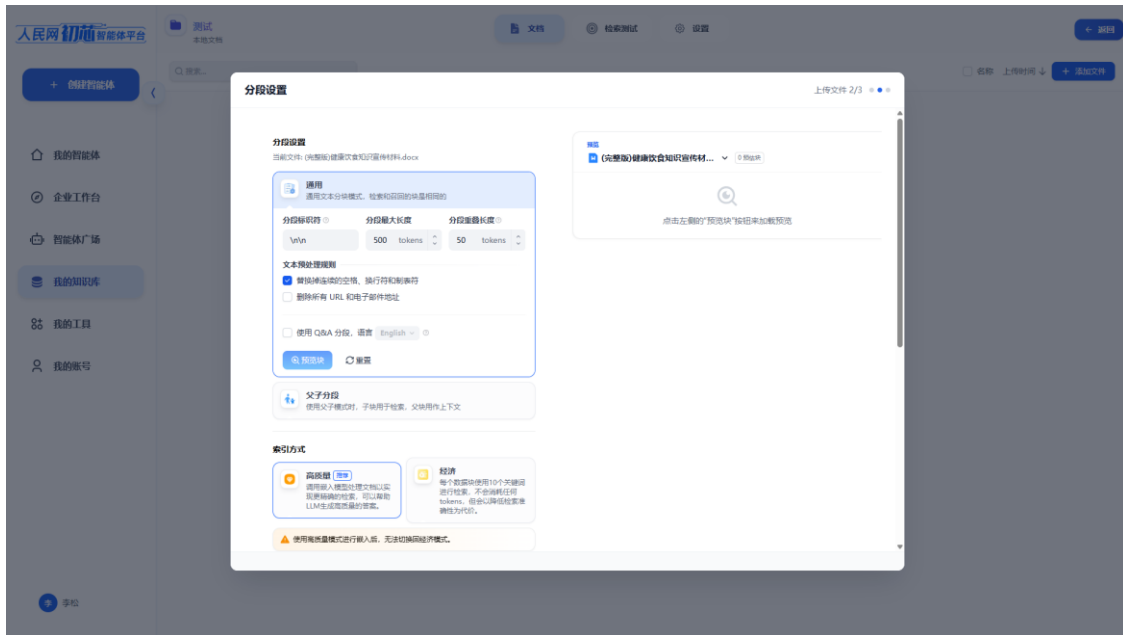
分段与清洗

分段：大语言模型存在有限的上下文窗口，通常需要将整段文本进行分段处理后，将与用户问题关联度最高的几个段落召回，即分段 TopK 召回模式。此外，在用户问题与文本分段进行语义匹配时，合适的分段大小将有助于匹配关联性最高的文本内容，减少信息噪音。

清洗：为了保证文本召回的效果，通常需要在将数据传入模型之前对其进行清理。例如，如果输出中存在不需要的字符或者空行，可能会影响问题回复的质量。为了帮助用户解决这个问题，人民网智能体开放平台提供了多种清洗方法，可以帮助用户在将输出发送到下游应用程序之前对其进行清理。

分段与清洗支持两种配置策略

- 自动模式
- 自定义模式



在自定义模式下，用户可以根据不同的文档格式和场景要求来配置文本的分段和清洗策略。

分段规则：

- 分段标识符，设置标识符如“\n”，系统将在文本中出现该标识符时分段；
- 分段最大长度，根据分段的文本字符数最大上限来进行分段，超出该长度时将强制分段；
- 分段重叠长度，设置分段间的重叠字符数，建议设置为分段长度的 10-25%，有助于保留分段之间的语义相关性，在多分段召回时提高召回效果。

预处理规则：

- 替换连续的空格、换行符和制表符；
- 删除所有 URL 和电子邮件地址；

索引方式

你需要选择文本的**索引方式**来指定数据的匹配方式，索引策略往往与检索方式相关，你需要根据场景需求来选择合适的索引方式。

高质量模式： 将调用 OpenAI 的嵌入接口进行处理，以在用户查询时提供更高的准确度。

经济模式： 会使用关键词索引方式，降低了准确度但无需花费 Token。

Q&A 模式： Q&A 分段模式功能，与上述普通的「Q to P」（问题匹配文本段落）匹配模式不同，它是采用「Q to Q」（问题匹配问题）匹配工作，在文档经过分段后，经过总结为每一个分段生成 Q&A 匹配对，当用户提问时，系统会找出与之最相似的问题，然后返回对应的分段作为答案。这种方式更加精确，因为它直接针对用户问题进行匹配，可以更准确地获取用户真正需要的信息。

在知识库上传文档时，系统将对文本进行分段，使得用户的提问（输入）能匹配到相关的文本段落（Q to P），最后输出结果。

检索设置

在高质量索引模式下，人民网智能体开放平台 提供了 3 种检索方案：

- **向量检索**，通过生成查询嵌入并查询与其向量表示最相似的文本分段。

- **全文检索**，索引文档中的所有词汇，从而允许用户查询任意词汇，并返回包含这些词汇的文本片段。
- **混合检索**，同时执行全文检索和向量检索，并附加重排序步骤，从两类查询结果中选择匹配用户问题的最佳结果，需配置 Rerank 模型 API。

三种检索方式的具体配置如下：

向量检索

定义：通过生成查询嵌入并查询与其向量表示最相似的文本分段。

TopK：用于筛选与用户问题相似度最高的文本片段。系统同时会根据选用模型上下文窗口大小动态调整片段数量。系统默认值为 3 。数值越高，预期被召回的文本分段数量越多。

Score 阈值：用于设置文本片段筛选的相似度阈值，即：只召回超过设置分数的文本片段。系统默认关闭该设置，即不会对召回的文本片段相似值过滤。打开后默认值为 0.5。数值越高，预期被召回的文本数量越少。

全文检索

定义：索引文档中的所有词汇，从而允许用户查询任意词汇，并返回包含这些词汇的文本片段。

TopK：用于筛选与用户问题相似度最高的文本片段。系统同时会根据选用模型上下文窗口大小动态调整片段数量。系统默认值为 3 。数值越高，预期被召回的文本分段数量越多。

Score 阈值：用于设置文本片段筛选的相似度阈值，即：只召回超过设置分数的文本片段。系统默认关闭该设置，即不会对召回的文本片段相似值过滤。打开后默认值为 0.5。数值越高，预期被召回的文本数量越少。

混合检索

定义：同时执行全文检索和向量检索，并应用重排序步骤，从两类查询结果中选择匹配用户问题的最佳结果。在此模式下可以指定“权重设置”（无需配置 Rerank 模型 API）或选择 Rerank 模型进行检索。

4.2 知识库及文档维护

查看文本分段

知识库内已上传的每个文档都会以文本分段（Chunks）的形式进行存储，你可以在分段列表内查看每一个分段的具体文本内容。

检查分段质量

文档分段对于知识库应用的问答效果有明显影响，在将知识库与应用关联之前，建议人工检查分段质量。

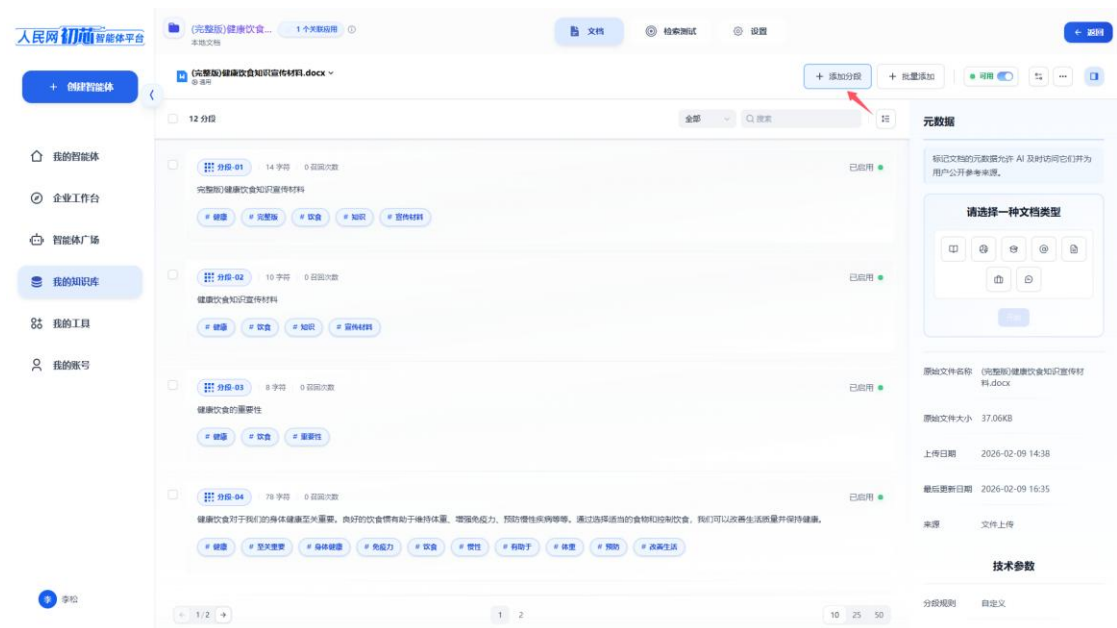
通过字符长度、标识符或者 NLP 语义分段等机器自动化的分段方式虽然能够显著减少大规模文本分段的工作量，但分段质量与不同文档格式的文本结构、前后文的语义联系都有关系，通过人工检查和订正可以有效弥补机器分段在语义识别方面的缺点。

检查分段质量时，一般需要关注以下几种情况：

- **过短的文本分段**，导致语义缺失；
- **过长的文本分段**，导致语义噪音影响匹配准确性；
- **明显的语义截断**，在使用最大分段长度限制时会出现强制性的语义截断，导致召回时缺失内容；

添加文本分段

在分段列表内点击 「 添加分段 」 ，可以在文档内自行添加一个或批量添加多个自定义分段。



批量添加分段时，你需要先下载 CSV 格式的分段上传模板，并按照模板格式在 Excel 内编辑所有的分段内容，再将 CSV 文件保存后上传。

编辑文本分段

在分段列表内，你可以对已添加的分段内容直接进行编辑修改。包括分段的文本内容和关键词。

添加文档

在「知识库 > 文档列表」点击「添加文件」，可以在已创建的知识库内上传新的文档。

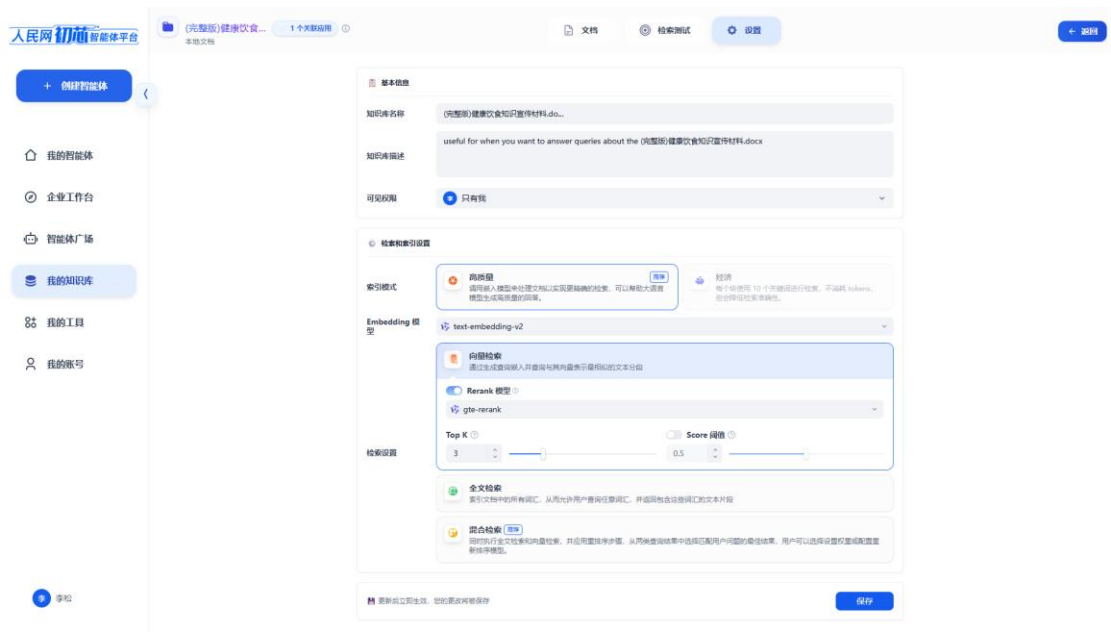
文档禁用和归档

禁用：数据集支持将暂时不想被索引的文档或分段进行禁用，在数据集文档列表，点击禁用按钮，则文档被禁用；也可以在文档详情，点击禁用按钮，禁用整个文档或某个分段，禁用的文档将不会被索引。禁用的文档点击启用，可以取消禁用。

归档：一些不再使用的旧文档数据，如果不想删除可以将它进行归档，归档后的数据就只能查看或删除，不可以进行编辑。在数据集文档列表，点击归档按钮，则文档被归档，也可以在文档详情，归档文档。归档的文档将不会被索引。归档的文档也可以点击撤销归档。

知识库设置

在知识库的导航中点击**设置**，你可以改变知识库的以下设置项：



知识库名称，定义一个名称用于识别一个知识库。

知识库描述，用于描述知识库内文档代表的信息

可见权限，可选择 「 只有我 」 或 「 所有团队成员 」，不具有权限的人将无法查阅和编辑数据集。

索引模式

检索设置

4.3 在应用内集成知识库

知识库可以作为外部知识提供给大语言模型用于精确回复用户问题，你可以在 人民网智能体开放平台 的所有应用类型内关 联已创建的知识库。

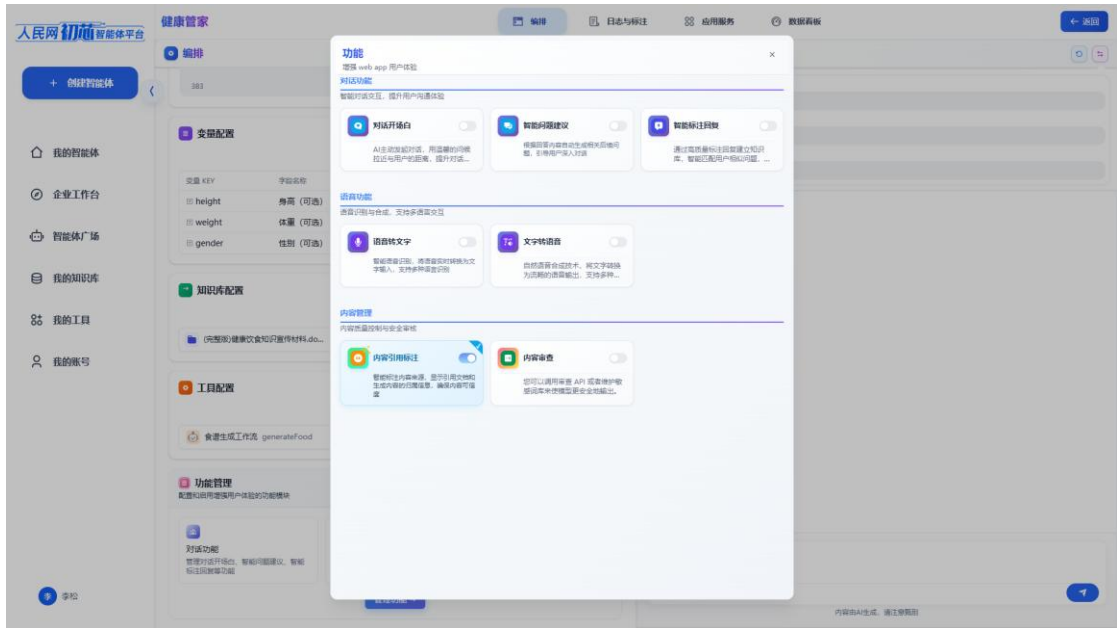
以Agent为例，使用流程如下：

1. 进入 **工作室 -- 创建应用 --创建Agent**
2. 进入 **上下文设置** 点击 **添加** 选择已创建的知识库
3. 在 **添加功能** 内打开 **引用和归属**
4. 在 **调试与预览** 内输入与知识库相关的用户问题进行调试
5. 调试完成之后**保存并发布**为一个 AI 知识库问答类应用

4.4 引用归属

引用与归属

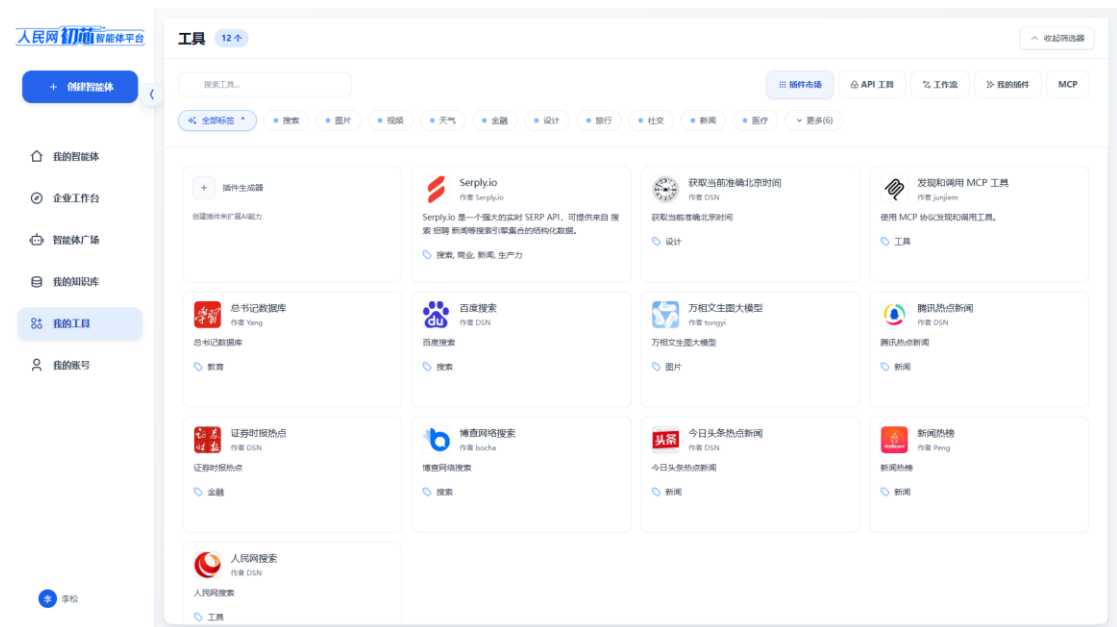
在应用内测试知识库效果时，你可以进入 **工作室 -- 添加功能 -- 引用归属**，打开引用归属功能。



开启功能后，当大语言模型回复问题时引用来自知识库的内容时，可以在回复内容下面查看到具体的引用段落信息，包括**原始分段文本**、**分段序号**、**匹配度**等。点击引用分段上方的跳转至知识库，可以快捷访问该分段所在的知识库分段列表，方便开发者进行调试编辑。

五、工具

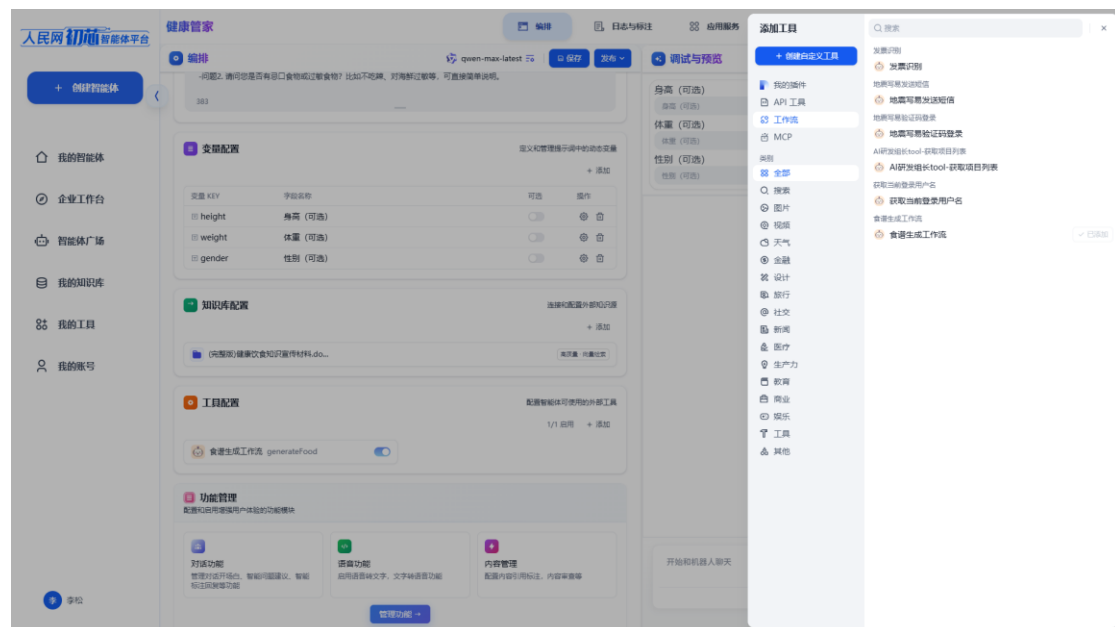
5.1 内置自定义工具



目前人民网智能体开放平台内置了可以使用的工具接口。

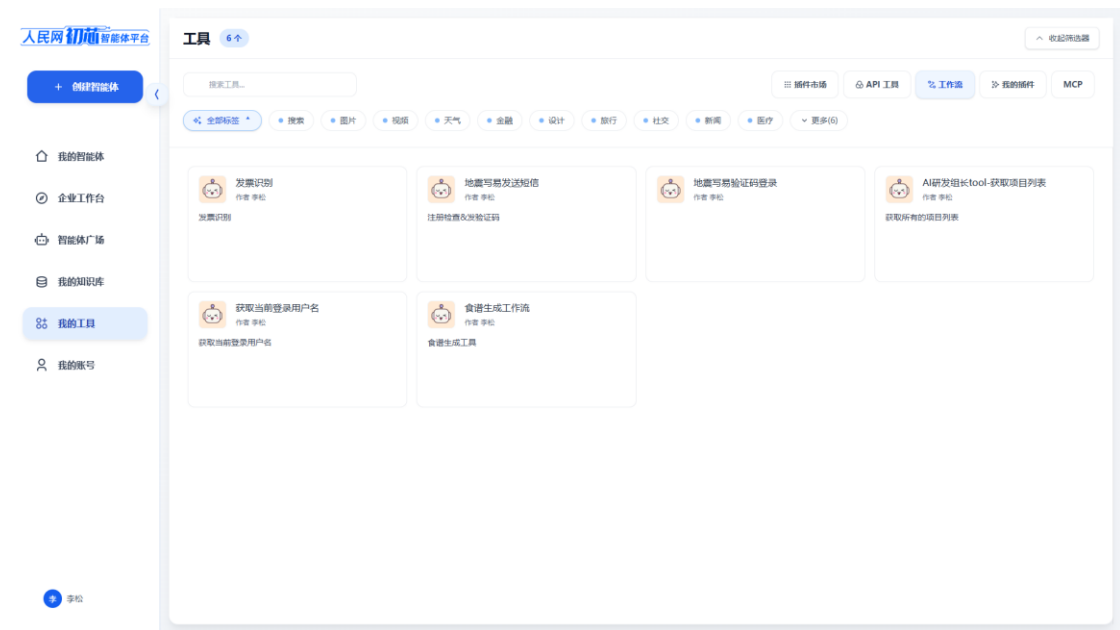
- Serply.io 搜索工具
- 北京时间
- MCP 调用工具
- 总书记数据库
- 百度搜索
- 万相文生图大模型
- 腾讯热点新闻
- 证券时报热点
- 博查网络搜索
- 今日头条热点新闻
- 新闻热榜
- 人民网搜索

如果想使用这些工具，在Agent编排页面中的工具模块，选择“添加”，在右侧的自定义栏目中选择合适主体的搜索接口工具。



5.2 工作流工具

发布为工具的工作流可以在Agent编排时使用，用户只能浏览和使用自己创建发布的工作流工具。



如果想使用工作流工具，在Agent编排页面中的工具模块，选择“添加”，在右侧的工作流栏目中选择合适主体的工作流工具。

