**南昌大学大学生创新创业训练计划项目**

**研 究 报 告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **项目名称** | **：** | | 多轮对话技术研究及在防洪减灾方面的应用 | |
| **项 目 级 别** | **：** | □国家级 □省级 ¢校级 | |
| **项 目 类 别** | **：** | ☑ 创新训练项目  □ 创业训练项目  □ 创业实践项目 | |
| **学院名称** | **：** | 软件学院 | |
| **项目组成员** | **：** | 李云瑞 彭武倜 黄涛 唐婧 | |
| **指导教师** | **：** | 曾长清 | |
| **资助金额** | **：** | 0元 | |
| **起 止 年 月** | **：** | 2022-09-30至2023-09-30 | |

**南昌大学教务处**

**二〇二贰叁年 九月** **摘要**

洪水灾害是由于江河、湖泊、水库水位急剧上升，大坝发生溢流或决口，水流入境等原因造成的。洪水除了造成重大农业灾害外，还造成工业损失乃至生命财产损失，是对人类生存起到威胁作用的十大自然灾害之一。

洪水灾害是我国发生频率最高、影响范围最广、对国民经济和社会经济发展影响最为严重的自然灾害，目前的防洪减灾系统都是基于传统方式开发的信息服务系统，很少和对话系统相结合，也很少采用图数据库技术来实现洪水传播时间的计算，少数地方开发了仅仅是简单的单轮对话，难以实用。对话系统依据对话的轮次可以分成单轮对话和多轮对话。单轮对话只会考虑当前轮次的信息，而多轮对话对人机对话的上下文、指代省略的补全和复杂需求的明确等问题都有着更为复杂的要求，也更加贴近现实生活中咨询、推荐、服务需求等应用场景。本项目主要研究的是任务型多轮对话系统，其是为了完成指定目标，如洪水传播时间的对话、降雨情况的时空分布等人机交互任务。

本项目先实现基于管道的任务型多轮对话系统，对意图识别和对话策略进行建模，使得对话系统具有很好的性能，然后基于图数据库建立洪水传播时间的数据库，并采用最短路径算法计算洪水传播时间，最后讲将二者结合起来，开发一个任务型多轮对话防洪减灾系统，从而提升防灾减灾系统的人机交互能力，为政府决策提供更好的服务。

**关键字：人工智能、多轮对话、深度学习、意图识别、自然语言理解**

**致谢**

通过这次“大学生创新创业训练计划”项目，让我们对所学的专业知识有了一个更为开拓创新的经历，同是也使得我们对相关学术领域和项目从无到有的运行有了一个全面的了解。在这次的“大学生创新创业训练计划”项目过程中，我们感受很多，收获很多。尤其是指导老师曾老师的敬业精神也令我更加感动，他始终如一的陪着我们，不厌其烦的为我们讲解，才使得我们的项目得以快速高效的完成。我们由衷的感谢在这次经历中的曾老师对我们给予的帮助。

**目录**

[第一章 研究导论 1](#_Toc25026)

[1.1 项目研究背景 1](#_Toc31752)

[1.2 问题陈述 2](#_Toc27064)

[1.3 项目研究意义 3](#_Toc2708)

[第二章 研究队伍 4](#_Toc19435)

[第三章 项目研究内容 5](#_Toc14103)

[3.1 项目研究内容 5](#_Toc22316)

[3.2 项目功能介绍 6](#_Toc17553)

[3.3 项目系统介绍 7](#_Toc31785)

[第四章 项目研究技术 9](#_Toc23542)

[4.1 技术路线 9](#_Toc22269)

[4.2 技术选型 10](#_Toc19469)

[4.3 技术框架 12](#_Toc23075)

[4.4 算法模型 14](#_Toc18694)

[第五章 项目研究过程 19](#_Toc19995)

[5.1 项目研究方法 19](#_Toc12813)

[5.2 项目环境配置 20](#_Toc13138)

[5.3 项目数据集 24](#_Toc21540)

[第六章 项目研究结果 34](#_Toc7425)

[6.1意图实体识别优化结果 37](#_Toc1487)

[第七章 项目分析与展示 43](#_Toc32001)

[7.1 技术难点 43](#_Toc8297)

[7.2 项目展示 44](#_Toc3981)

[第八章 参考资料 54](#_Toc16442)

# 第一章 研究导论

## 1.1 项目研究背景

随着气候变化和自然灾害频发，洪水灾害成为世界各地面临的严峻挑战之一。传统的防洪减灾方法在信息传递、决策支持和公众参与方面存在不足。为了应对这一挑战，本项目旨在结合多轮对话技术和图数据库技术，开发一个任务型多轮对话防洪减灾系统，以提升防灾减灾系统的人机交互能力，为政府决策提供更好的服务。相比于传统的防洪减灾预测系统，运用多轮对话技术有以下优势：

1. **单轮对话和多轮对话**：多轮对话系统较单轮对话系统更适合处理复杂任务，具有广泛的应用前景。本项目聚焦于任务型多轮对话系统，以满足防洪减灾领域的需求。
2. **基于管道的任务型对话系统**：基于管道的方法具有模块化的优势，允许灵活组合各个子模块以构建完整的对话系统。项目将采用这种方法，分别考虑自然语言理解、对话管理和自然语言生成模块。
3. **防洪减灾决策支持系统**：现有的防洪减灾决策支持系统尚未充分融合多轮对话技术和图数据库计算，因此存在改进空间。该项目的目标是填补这一技术鸿沟，以提供更全面、实用且高效的决策支持。

本项目旨在将多轮对话技术与图数据库技术相结合，开发一种创新的防洪减灾决策支持系统，以应对日益频发的洪水灾害，提高公众安全并为政府决策提供强有力的工具。

## 1.2 问题陈述

在基于多轮对话技术在防洪减灾的应用中，有许多对话框架可以选择，本项目组最终使用的框架为Rasa框架，但是在实际运用中还是存在以下问题：

1. 意图实体识别不准确
2. 槽填充对模型意图拟合概率产生影响
3. 数据库与rasa项目相结合
4. 在前端网页对话窗口输出图表
5. 训练数据格式要求
6. 构建中文对话机器人

## 1.3 项目研究意义

传统洪水预测项目主要依赖于物理模型、气象数据、水文数据等科学方法来估计洪水的发生和影响。这些方法通常基于历史数据和数学方程，计算出洪水可能的发生概率和程度。我们的项目引入了多轮对话技术和图数据库，采用更智能的方法来实现洪水预测。它能够实时与用户互动，收集关于洪水情况的信息，并利用图数据库来计算洪水传播时间，提供更准确的洪水预测。

1. **提升防洪减灾效率**：该项目的研究意义首先在于提高防洪减灾系统的效率。通过开发一个任务型多轮对话系统，能够更快速地与公众互动、收集信息，并提供实时的洪水相关数据，从而加速决策制定和响应行动的速度，有助于减轻洪水带来的损害。
2. **加强人机交互**：多轮对话系统的引入将加强防洪减灾系统与用户之间的互动。这不仅提高了系统的可用性，还使得用户能够更容易地获得所需信息和支持。这对于公众的安全意识和应对洪水事件的能力至关重要。
3. **提供个性化服务**：基于多轮对话的系统能够根据用户的需求和上下文提供个性化的服务。这意味着系统可以为不同地区和不同用户提供定制化的建议和信息，更好地满足他们的特定需求。
4. **优化洪水传播时间计算**：引入图数据库和最短路径算法用于洪水传播时间计算，将使得这一关键信息更加准确和可靠。这对于决策制定和资源调度具有重要意义，有助于最大程度地减少洪水灾害的影响。
5. **跨学科研究贡献**：本项目融合了多个领域的技术和知识，包括自然语言处理、数据库管理、地理信息系统等。这将为跨学科研究提供宝贵的实践经验，并促进不同领域之间的合作和知识交流。
6. **应对气候变化挑战**：随着气候变化的不断加剧，洪水事件可能会变得更加频繁和复杂。因此，提高防洪减灾系统的效能和智能化至关重要。该项目的研究意义也在于为应对气候变化带来的挑战提供了一种创新的方法。

综上所述，本项目具有重要的社会和科学意义，不仅有助于提高防洪减灾效率，还为人机交互、个性化服务、跨学科研究以及气候变化适应提供了有力支持，为社会的可持续发展和公众的安全提供了有力的保障。

# 第二章 研究队伍

|  |  |  |
| --- | --- | --- |
| 姓名 | 主要负责事项 | 细致事项 |
| 李云瑞 | 项目负责人、系统开发人员、系统测试 | 编制项目总体规划和研究方案，完成rasa后台action和对话逻辑的编写，以及撰写作品文档和相关报告，项目PPT。 |
| 彭武倜 | 系统开发人员、系统测试  数据库设计员、管理员 | 完善项目总体规划和研究方案，  编写完善rasa训练数据和action。  完成中文管道的设置。完成数据库的设置并负责日常维护和管理，撰写报告文档。 |
| 唐婧 | 系统开发人员 | 根据项目需求设计对应前端ui，实现前端界面与rasa后台交互，并在前端构建相应图表，以及撰写论文和相关文档。 |
| 黄涛 | 系统测试、算法研发人员 | 测试系统，完成水位流量关系线生成 |

# 第三章 项目研究内容

## 3.1 项目研究内容

* （1）提出基于外部记忆的意图识别模型。多轮对话更多的是考虑历史信息记录，使用循环神经网络来进行意图识别比较适合。随着人机对话的复杂性增加，循环神经网络因为内部记忆单元存储容量太小，容易丢失关键信息，所以需要设计基于外部记忆的意图识别模型。此模型是以BiLSTM为基础，同时引入外部记忆单元和自注意力机制，最后添加Dropout防止神经网络出现过拟合。
* （2）使用DQN算法实现对话策略模块。在传统基于最大似然估计模型中每一轮对话中所获得的回复是对话系统生成概率最大的句子，很多万能回复容易被选中，当系统和用户出现相似或者相同的语句的时候，系统容易陷入死循环；对话系统一般有很大的状态空间或动作空间，对话策略使用的Q-Learning要想建立一个Q表去维护要消耗大量的内存。为了解决上述问题在对话策略中引入DQN算法，其是计算的是每个回复语句能够为对话系统中输入带来多少的未来奖励，进而来挑选能够为系统带来最大未来奖励的语句。
* （3）设计实现河流的防洪指标知识图谱，包括：图数据库建设；知识图谱可视化；河道站查询；洪水传播时间查询；水位流量关系曲线查询，降水的时空分布查询。
* （4）实现基于任务型多轮对话的防洪减灾系统。重点对将前面提出的模型并结合相关算法应用到防灾减灾服务的对话上，最后很好的为政府防灾减灾决策提供支持。

本项目基于rasa框架构建，rasa框架的核心技术算法有**条件随机场（CRF）,** **词嵌入技术（Word Embeddings）,自然语言理解（NLU）,**我们对这些rasa的核心技术进行了具体的研究，内容如下:

**一、条件随机场（CRF）**

Rasa使用条件随机场（Conditional Random Fields，CRF）作为一种核心的序列标注算法，用于实体识别（Entity Recognition）任务。CRF是一种经典的统计建模技术，常用于序列标注问题，例如自然语言处理中的实体识别。

1. **序列标注问题**：在自然语言处理中，序列标注问题是指给定一个输入序列（通常是一段文本），要求标注出其中的子序列，即找到文本中的实体，如日期、地点、组织名称等。CRF在这种问题中非常有效，因为它能够考虑到标签之间的依赖关系。
2. **特征工程**：在CRF中，需要定义一组特征来描述输入文本中的每个单词或标记。这些特征可以包括单词本身、词性、上下文信息等。Rasa会从文本中提取这些特征，以便将文本映射到标签序列。
3. **标签定义**：在实体识别任务中，需要定义一组目标标签，用于表示不同类型的实体。例如，可以定义"DATE"标签用于识别日期实体，"LOCATION"标签用于识别地点实体，等等。
4. **模型训练**：Rasa使用CRF算法来训练实体识别模型。训练过程中，模型学习如何为每个输入标记分配正确的标签，以最大化训练数据的似然性。
5. **标签传递**：CRF模型通常考虑了标签之间的依赖关系，这意味着它会考虑到前一个标记对当前标记的影响。这有助于更准确地识别实体，尤其是在实体之间存在关联的情况下。
6. **评估和调优**：一旦模型训练完成，可以使用评估数据集来评估其性能。通常使用指标如精确度、召回率、F1分数等来衡量模型的准确性。根据评估结果，可以对模型进行调优以提高性能。

Rasa中的CRF是用于实体识别的一种强大的统计建模技术。它允许Rasa识别输入文本中的实体，并将其映射到预定义的实体类型，从而有助于理解用户输入和提供更准确的响应。CRF考虑了标签之间的依赖关系，这使得它在处理序列标注问题时非常有效。

二、**词嵌入技术（Word Embeddings）**

词嵌入技术（Word Embeddings）是一种将单词表示为连续向量的自然语言处理（NLP）技术。这些连续向量在NLP任务中非常有用，因为它们可以将离散的文本数据转换为连续的数值表示，从而使计算机能够更好地理解和处理文本数据。以下是有关词嵌入技术的重要信息：

1. **连续向量表示**：在传统的文本处理中，单词通常被表示为离散的符号。然而，这种表示方式难以捕捉单词之间的语义关系和上下文信息。词嵌入技术通过将每个单词映射到一个连续的向量空间中，使得每个单词都有一个具体的数值向量表示。
2. **分布式表示**：词嵌入的核心思想是“相似的词在向量空间中应该具有相似的表示”。这意味着如果两个单词在语义上相似，它们的词嵌入向量在空间中会更加接近。这种分布式表示有助于捕捉单词之间的语义关系，例如，可以通过向量运算计算出"king - man + woman"得到"queen"。
3. **训练方法**：词嵌入模型通常是通过大规模文本语料库进行训练的。常见的词嵌入模型包括Word2Vec、GloVe（Global Vectors for Word Representation）、FastText等。这些模型根据上下文单词的共现关系来学习单词的嵌入向量。
4. **词嵌入引擎**：Rasa支持多种词嵌入引擎，包括spacy和BERT等。这些引擎用于将单词映射到连续的向量空间中，以便在NLP任务中使用。本项目最终使用的就是spacy引擎。
5. **预训练的词嵌入**：现在，有许多预训练的词嵌入模型可供使用，如Word2Vec、GloVe和BERT。这些模型在大规模文本数据上进行了训练，并提供了通用的词嵌入表示，可以用于各种NLP任务，而无需从头开始训练模型。

词嵌入技术是NLP领域中的一个重要进展，它使计算机能够更好地理解和处理文本数据，有助于提高NLP任务的性能。通过将单词表示为连续向量，词嵌入技术有助于捕捉语义关系，从而更好地理解单词的含义和上下文。

三、**自然语言理解（NLU）**

Rasa在其对话系统中使用自然语言理解（Natural Language Understanding，NLU）来解析和理解用户的输入文本。以下是Rasa中使用NLU的一般流程和主要组件：

1. **文本输入**：
   * 用户与Rasa对话系统进行交互，输入自然语言文本，这可以是问题、请求或指令。
2. **NLU管道**：
   * Rasa的NLU组件通常构建一个处理管道，包括以下步骤：

**分词（Tokenization）**：将输入文本分解为单词或标记的步骤。这有助于将长文本划分为更小的单元，以便进一步处理。

* + - **词干化（Stemming）和词形还原（Lemmatization）**：对单词进行规范化处理，将它们转化为标准形式，以减少词汇多样性的影响。
    - **停用词去除（Stop Word Removal）**：去除文本中的停用词，这些词汇通常是常见且不携带重要语义信息的词汇。
    - **词嵌入（Word Embeddings）**：将单词映射到连续的向量空间，以便更好地理解词汇的语义。Rasa支持多种词嵌入引擎，如spaCy和BERT。
    - **意图分类（Intent Classification）**：将用户输入的文本分类为不同的意图，这是对话系统中的核心任务之一。NLU组件使用机器学习模型将文本映射到预定义的意图类别。
    - **实体识别（Entity Recognition）**：识别和标记用户输入文本中的实体，如日期、地点、产品名称等。这有助于系统更好地理解用户的需求和信息。

1. **意图和实体输出**：
   * NLU管道解析用户输入并输出两个主要部分：
     + **意图（Intent）**：确定用户输入的主要目的或意图。例如，用户可能表达了“订票”的意图。
     + **实体（Entities）**：标识和提取用户输入中的实体信息。例如，用户可能提到了日期、目的地和乘客人数等实体信息。
2. **意图和实体传递**：
   * NLU输出的意图和实体信息将传递给对话管理（Dialogue Management）组件，以便对话系统可以根据用户的输入执行适当的响应行动。

Rasa使用NLU来解析用户的自然语言输入，将其转化为可理解的结构化数据（意图和实体），以支持对话系统的自动化响应。这使得Rasa能够更好地理解用户的意图和需求，从而提供更智能的对话体验。开发者可以根据项目需求自定义和配置NLU管道，以优化对话系统的性能。

## 3.2 项目功能介绍

**1. 人机对话窗口：**

* 提供用户友好的人机对话窗口，允许用户通过自然语言与系统进行交互。
* 用户可以通过对话窗口提出问题、请求信息，系统将自动回应并提供相应的解答或操作。

**2. 系统登录：**

* 支持用户身份验证和登录功能，确保只有授权用户才能访问系统。
* 用户可以使用安全的登录凭据，如用户名和密码，以访问系统的高级功能。

**3. 水位流量关系线：**

* 提供水位与流量之间的关系图示，允许用户了解不同水位下的水流情况。
* 用户可以查看水位和流量的关系，以更好地理解洪水情况。

**4. 知识图谱：**

* 构建知识图谱，整合了有关洪水、气象、地理信息等方面的知识。
* 用户可以查询和浏览知识图谱，获取与决策支持相关的信息。

**5. 洪水传播时间：**

* 开发洪水传播时间计算模块，可以根据地理位置和洪水条件预测洪水的传播时间。
* 用户可以输入特定位置和洪水条件，系统将计算并返回洪水到达该地点的预计时间。

**6. 河道站查询：**

* 提供河道站点查询功能，用户可以查找特定河流或水域的监测站点。
* 用户可以获得有关监测站点的信息，如名称、位置、实时数据等。

**7. 降水查询：**

* 具备降水数据查询模块，用户可以获取特定地区的降水信息。
* 用户可以查询降雨量的历史记录、实时数据以及预测数据，以便更好地了解降雨情况。

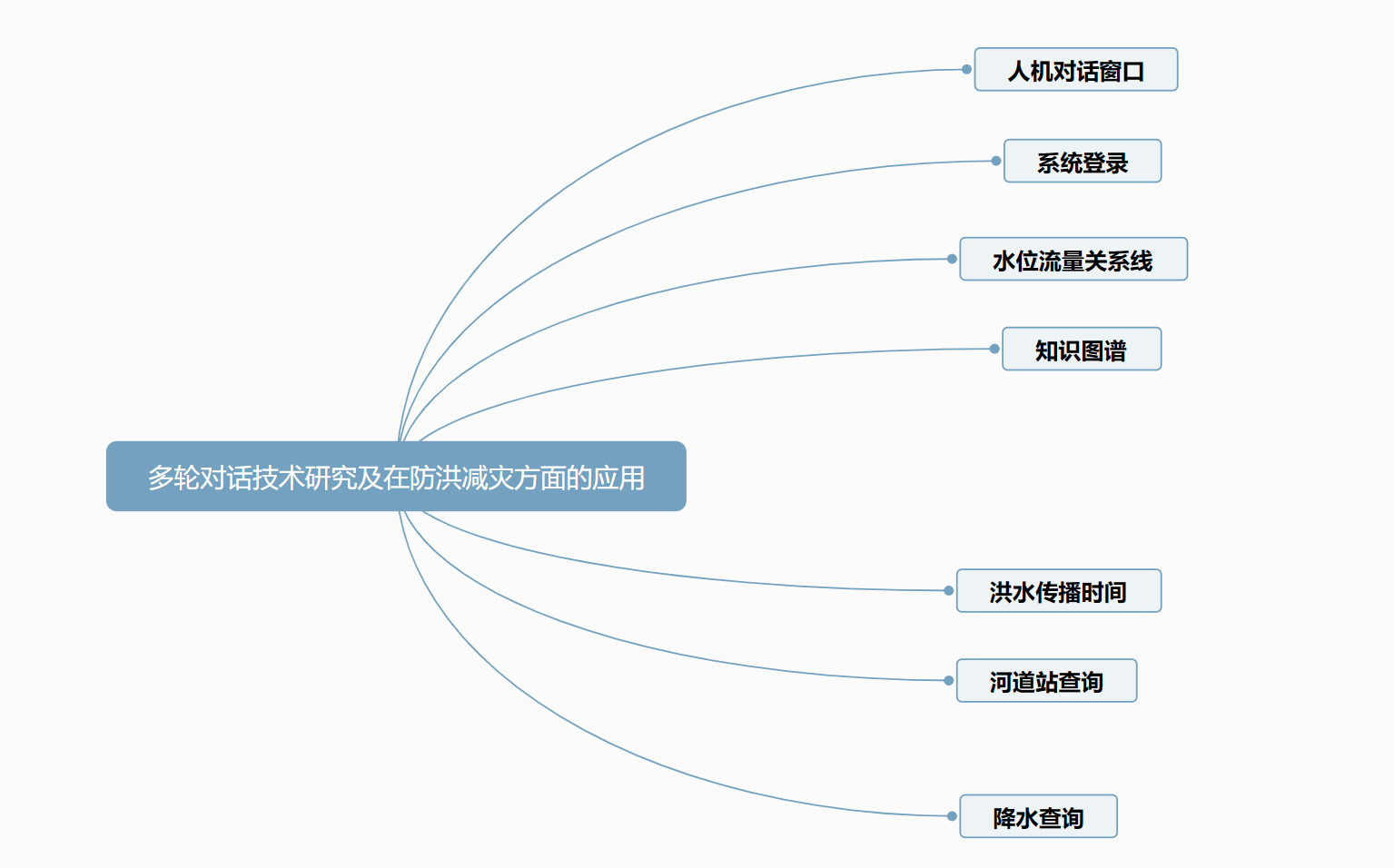


图 3-1 项目功能

## 3.3 项目系统介绍

1、河道站数据查询子系统

（1）河道站数据库

（2）Rasa意图识别

（3）Rasa人机对话逻辑

（4）后台捕获用户对话中的实体

（5）利用获取的信息与数据库进行交互

2、对话窗口子系统

（1）用户登录

（2）用户与防洪减灾机器人交互

（3）河道站知识图谱查看

（4）水位流量关系线查看

3、洪水传播时间预测子系统

（1）河道站洪水传播时间图数据库

（2）Rasa意图识别

（3）Rasa form的触发

（4）Rasa人机对话逻辑

（5）后台捕获用户对话中的实体

（6）利用获取的信息与数据库进行交互

（7）计算得出洪水传播时间

图 3-2 系统总体框架图

# 第四章 项目研究技术

## 4.1 技术路线

**需求分析**

1、定义项目的主要目标和需求，包括对话系统功能、数据需求和用户交互等方面的需求。

2、收集相关领域的数据和信息，以了解防洪减灾的相关背景和要求。

**系统设计**

1、设计对话系统的架构，包括自然语言理解（NLU）、对话管理（Dialogue Management）和自然语言生成（NLG）模块。

2、确定系统的输入和输出接口，以及与用户的交互方式，如人机对话窗口和系统登录。

**数据准备**

1、收集和整理洪水传播时间的数据，包括水位、流量、地理信息等相关数据。

2、构建知识图谱，整合地理信息、水文信息等数据，以支持系统的智能决策。

3、准备降水数据，以了解降雨情况的时空分布。

**模型开发**

1、开发自然语言理解（NLU）模块，使用Rasa框架中的DIETClassifier等工具来识别用户意图和提取命名实体。

2、开发对话管理（Dialogue Management）模块，定义对话策略和系统行为，以响应用户的请求和问题。

3、实现自然语言生成（NLG）模块，以生成自然语言文本作为系统响应。

**集成与测试**

1、集成多轮对话系统到人机对话窗口和系统登录接口，以提供用户访问和交互的方式。

2、集成水位流量关系线和知识图谱，以支持系统的智能决策。

3、集成洪水传播时间计算模块，使用最短路径算法等方法计算洪水传播时间。

4、集成河道站查询和降水查询功能，以满足用户的信息需求。

5、进行系统测试和调试，确保系统在各种情况下都能正常工作。

**优化和改进**

1、对系统进行性能评估和用户反馈收集，以识别问题并进行改进。

2、根据用户反馈和需求，不断优化对话系统的功能和性能。

**部署与维护**

1、部署防洪减灾决策支持系统，确保其在实际应用中稳定运行。

2、提供培训和文档，以帮助系统管理员和维护人员了解系统的操作和维护。

3、持续监控系统性能，定期更新数据和模型，以适应新的洪水和气象数据。

## 4.2 技术选型

**4.2.1 基于SpaCy中文NLP库构建中文对话机器人**

中文实体识别（Named Entity Recognition，NER）是自然语言处理领域的一个重要任务，其目标是从文本中识别和分类命名实体，如人名、地名、组织名、日期、时间、货币、百分比等。这些实体通常是具有特定语义含义的词汇或短语。

**1. 词语划分（Word Segmentation）：**

* 中文是一种分字语言，词语之间通常没有明显的空格或分隔符。因此，首先需要进行分词，将连续的字符序列划分为单词或词语，以便识别实体。分词错误可能导致实体边界的错误。

**2. 实体类别和多样性（Entity Types and Diversity）：**

* 中文包含多种类型的实体，包括人名、地名、组织名、日期、时间、货币等，因此需要具备多种实体类别的分类模型。此外，中文实体识别可能涉及大量的地名，而地名通常没有明显的上下文信息。

**3. 字符编码（Character Encoding）：**

* 中文字符编码通常更加复杂，需要使用Unicode等编码标准来表示中文字符。这可能会涉及字符集合的处理和转换。

**4. 歧义性（Ambiguity）：**

* 由于中文词语较短，上下文信息有时不足以消除实体的歧义。例如，"华为"既可以是一个公司名，也可以是一个人名。

SpaCyNLP相对于其他NLP库和工具具有多个优势，使其成为自然语言处理（NLP）任务的流行选择之一。以下是使用SpaCyNLP相对于其他NLP库的主要优势：

1. **性能和速度优越：** SpaCyNLP以其卓越的性能和速度而闻名。它经过优化，具有高度的效率，特别适用于大规模文本处理。相对于其他一些NLP库，SpaCy的处理速度更快，这使其在实时或高吞吐量应用中具有竞争优势。
2. **预训练语言模型：** SpaCy提供了多种多样的预训练语言模型，包括英语、德语、法语、西班牙语、中文等多种语言。这些预训练模型包括分词、词性标注、命名实体识别、依赖分析等多种NLP任务，可以用于快速实现各种自然语言处理任务。
3. **简单易用的API：** SpaCy的API设计非常简洁和用户友好，使得开发者可以轻松地进行文本处理和NLP任务。其文档详细且清晰，对于新手来说很容易上手。
4. **多语言支持：** SpaCy支持多种语言，包括主要的世界语言以及一些较少使用的语言。这使得它成为跨文化和跨语言NLP任务的理想选择。
5. **模块化和可扩展性：** SpaCy的架构非常模块化，可以轻松添加自定义组件和管道。这使得开发者可以根据具体任务的需求扩展和定制NLP流水线。
6. **丰富的特性：** SpaCy不仅提供了基本的文本处理功能，还包括了命名实体识别、词性标注、依赖分析、实体链接、文本分类、词向量表示等丰富的NLP特性，可以用于多种任务。
7. **社区支持和生态系统：** SpaCy有一个强大的社区，提供了丰富的资源、插件和扩展，以满足不同领域和应用的需求。这个生态系统使得在不同项目中共享和复用NLP组件变得更加容易。
8. **开源和免费：** SpaCy是开源软件，可以免费使用和修改，适用于商业和研究项目。

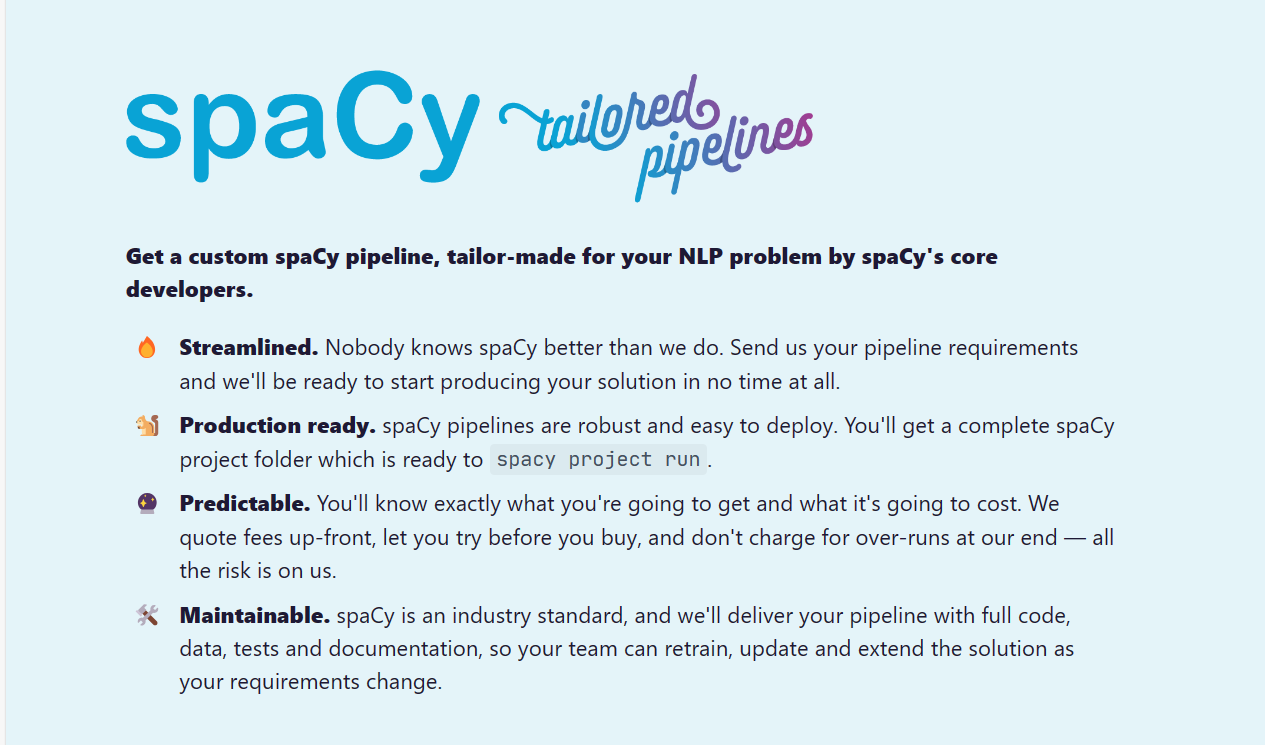


图4-2-1 SpaCy NLP的优势

**4.2.2 基于Vue2构建前端交互界面**

Vue.js是一个用于构建用户界面的渐进式JavaScript框架。它允许开发者通过组件化方式构建交互式的Web应用程序。Vue2是Vue.js的较早版本，但仍然受到广泛支持和使用。它有如下优势：

**1.组件化开发：**Vue2鼓励将界面划分为多个可重用的组件。每个组件都包含了自己的HTML、 CSS和JavaScript，使得开发更加模块化和可维护。这种组件化开发风格有助于构建复杂的用户 界面。

**2.数据驱动：**Vue2采用了数据驱动的思想，开发者只需关注数据的状态变化，界面会自动更新 以反映这些变化。这降低了手动DOM操作的需求，提高了代码的可读性和可维护性。

**3.指令和事件处理：**Vue2提供了一系列指令，如v-bind、v-model、v-for等，用于处理DOM 元素和数据的绑定。它还允许开发者声明性地处理事件，如@click、@input等。

**4.路由管理：**对于单页应用（SPA），Vue2可以与Vue Router结合使用，实现客户端路由管理。 这使得在应用内切换视图成为可能，而不需要重新加载整个页面。

**5.状态管理：**对于大型应用程序，Vue2可以与Vuex结合使用，实现全局状态管理。这使得多个 组件之间可以共享数据，而不需要通过逐层传递props。

**6.生态系统：**Vue2拥有丰富的生态系统，包括大量的第三方插件和库，可用于增强开发体验和 功能扩展。例如，可以使用Vue CLI来快速搭建Vue2项目，或使用Element UI来创建漂亮的界 面组件。

**7.学习资源：**学习Vue2的过程非常容易，因为它拥有详细的文档和丰富的在线教程。开发者可以轻松上手，并逐渐掌握Vue2的高级特性。



图4-2-2 Vue

**4.2.3 基于ElementUI快速构建前端界面**

Element UI是一个流行的Vue.js组件库，提供了一系列现成的UI组件，可帮助开发者快速构建漂亮的前端用户界面。以下是基于Element UI构建前端界面的关键要点：

**1.Element UI简介**：Element UI是一个基于Vue.js的前端组件库，旨在提 供现成的UI组件，帮助开发者轻松构建现代Web应用程序的用户界面。它 具有丰富的组件，如按钮、表格、表单、对话框等，可用于创建各种界面元 素。

**2.快速搭建**：Element UI的组件可以直接集成到Vue2项目中，无需从头开 始编写UI元素。这使得构建前端界面变得非常高效，特别是对于需要快速 开发原型或MVP的项目。

**3.丰富的组件**：Element UI提供了多种UI组件，覆盖了各种需求。例如， 可以使用<el-button>组件创建按钮，使用<el-table>组件展示数据表格， 使用<el-form>组件构建表单等。这些组件都具有灵活的配置选项，可根据 项目需求进行定制。

**4.响应式设计**：Element UI的组件都经过响应式设计，适应不同屏幕尺寸和 设备。这确保了您的应用程序在各种环境中都能提供出色的用户体验。

**5.主题定制**：Element UI允许开发者根据项目的视觉风格定制主题。通过调 整颜色、字体和其他样式属性，可以轻松使界面与品牌一致。

**6.国际化支持**：对于国际化项目，Element UI提供了多语言支持，使您可以 轻松地将应用程序本地化为不同的语言和地区。

**7.社区支持**：Element UI有一个活跃的社区，提供了大量的文档、示例和插 件，以帮助开发者解决问题并扩展功能。这个社区支持使得在项目中使用 Element UI变得更加容易。

**8.学习资源**：学习如何使用Element UI非常简单，因为它具有清晰的文档 和示例。开发者可以快速上手，构建出具有专业外观和良好交互性的前端界 面。

****

图4-2-3 Element

**4.2.4 基于axios实现异步交互**

在构建现代Web应用程序时，与后端服务器进行异步交互是不可或缺的一部分。axios是一个流行的JavaScript库，用于发起HTTP请求，并且非常适合在前端项目中实现异步通信。以下是关于基于axios实现异步交互的关键要点：

**1.axios简介：**axios是一个基于Promise的HTTP客户端，可用于在浏览器 和Node.js中进行HTTP请求。它提供了简洁的API，使得发送GET、POST、 PUT、DELETE等请求变得非常容易。

**2.异步请求：**axios允许前端应用程序发起异步HTTP请求，与后端服务器通 信。这是获取数据、发送表单、执行CRUD操作等任务的常见方式。

**3.Promise风格**：axios的API使用Promise风格的语法，这意味着您可以 使用.then()和.catch()来处理异步请求的响应和错误。这种风格的代码更 加清晰和可维护。

**4.拦截器：**axios允许您在请求和响应之前应用拦截器，以执行预处理或后 处理操作。这对于添加认证头、处理错误、或数据转换非常有用。

**5.并发请求：**axios支持并发请求，您可以同时发起多个请求，并等待它们 全部完成后执行特定的操作。

**6.错误处理：**axios具有强大的错误处理机制，可以捕获网络错误、HTTP错 误状态码以及应用程序级别的错误。这有助于提供友好的用户反馈和调试应 用程序。

**7.取消请求：**在某些情况下，您可能需要取消正在进行的HTTP请求，以避 免不必要的资源消耗。axios允许您创建可取消的请求，提高了性能和用户 体验。

**8.跨域请求：**对于跨域请求，axios可以轻松处理CORS（跨域资源共享）设 置，并支持跨域Cookie传递等功能。

**9.与Vue2集成：**axios可以轻松与Vue2集成，以在Vue组件中进行异步请 求。这使得在前端界面中获取和展示数据变得非常简单。

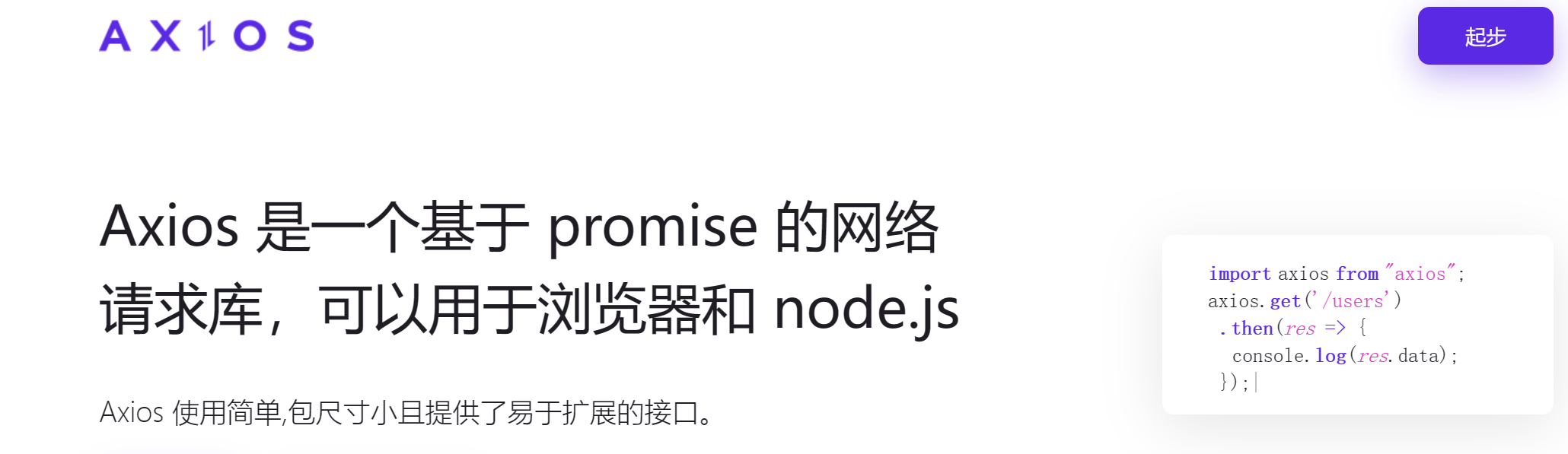


图4-2-4 Axios

**4.2.5 基于Flask构建后端**

Flask是一个Python Web框架，专注于简单性和轻量级性能。它是构建Web后端的理想选择，特别适合小型到中型的应用程序。以下是关于基于Flask构建后端的关键要点：

**1.Flask简介：**Flask是一个微型的、轻量级的Web框架，专注于提供基 本的工具和组件，以帮助开发者构建Web应用程序。它采用了"微"的设 计哲学，允许开发者选择并集成所需的功能。

**2.路由和视图函数：**在Flask中，您可以定义路由和视图函数，将URL 映射到特定的处理函数。这使得构建不同端点的API非常简单，您可以 根据请求的URL执行不同的操作。

**3.模板引擎：**Flask集成了Jinja2模板引擎，使得渲染动态HTML页面 变 得容易。您可以使用模板引擎将数据注入HTML模板，以动态生成页面 内 容。

**4.RESTful支持：**Flask天生支持RESTful API的构建。通过定义不同的HTTP 方法（GET、POST、PUT、DELETE等）和路由，您可以创建符合REST原则的 API。

**5.数据库集成：**Flask可以与各种数据库（如SQLite、MySQL、PostgreSQL 等）集成，使用ORM（对象关系映射）库（如SQLAlchemy）来管理数据。

**6.中间件：**Flask支持中间件，允许您在请求到达视图函数之前或之后执行 一些处理。这对于身份验证、日志记录和请求处理非常有用。

**7.扩展：**Flask具有丰富的扩展库，可用于添加额外的功能，如身份验证、 CSRF保护、API文档生成等。这些扩展简化了常见任务的实现。

**8.部署选项：**Flask可以轻松部署到各种Web服务器上，如Gunicorn、uWSGI、 Apache等。您可以选择最适合您应用程序的部署选项。

**9.学习资源：**Flask拥有广泛的文档和社区支持，使得学习和使用它变得容 易。无论是初学者还是有经验的开发者，都可以受益于Flask的简单性和灵 活性。

****

图4-2-5 Flask

**4.2.6 使用Docker构建封装和管理Rasa和其依赖项**

Docker是一种容器化平台，用于轻松地打包、分发和运行应用程序及其依赖项。容器是一种轻量级的虚拟化技术，允许您在单个容器中打包应用程序和其所有依赖项，包括操作系统、库、配置文件和运行时环境。Docker的主要目标是提供一种一致的环境，使开发、测试和部署应用程序变得更加可靠和可重复。

**1.Docker简介：**Docker 是一种开源的容器化平台，用于构建、部署和管理应用程序和服务。它的核心概念是容器，容器是一种轻量级、独立于环境的执行单元，包括应用程序和其依赖项。

**2.容器化技术：**Docker 使用容器化技术，将应用程序及其依赖项封装到一个独立的容器中。这个容器包含了应用程序的代码、运行时环境、库和配置，使其能够在不同的环境中以一致的方式运行，无论是开发、测试还是生产环境。

**3.镜像：**Docker 镜像是容器的构建模块。它包含了应用程序的文件系统、运行时环境和配置信息。镜像是不可修改的，一旦创建就不能更改，但可以基于现有镜像创建新的镜像。

**4.容器：**容器是基于镜像创建的运行实例。容器可以启动、停止、删除，并且它们相互隔离，不会互相干扰。每个容器都有自己的文件系统、网络和进程空间，但共享主机操作系统的内核。

**5.Docker Hub：**Docker Hub 是一个公共的 Docker 镜像仓库，开发人员可以在其中分享和获取 Docker 镜像。它包含了大量的官方和社区维护的镜像，用于各种不同的应用程序和服务。

**6.Dockerfile：**Dockerfile 是一个文本文件，用于定义如何构建 Docker 镜像。它包含了一系列指令，例如基于哪个基础镜像、如何添加文件、设置环境变量等。通过 Dockerfile，可以自动化地构建镜像，确保一致性和可重复性。

**7.** **Docker Compose：**Docker Compose 是一个工具，用于定义和管理多个 Docker 容器的应用程序。通过 Compose 文件，可以轻松定义容器之间的关系、网络设置和环境变量，以便将多个容器协同运行。

**8.** **Docker Swarm：**Docker Swarm 是 Docker 提供的容器编排工具，用于管理多个容器的集群。它允许自动化容器的部署、扩展和负载均衡，以确保高可用性和容错性。

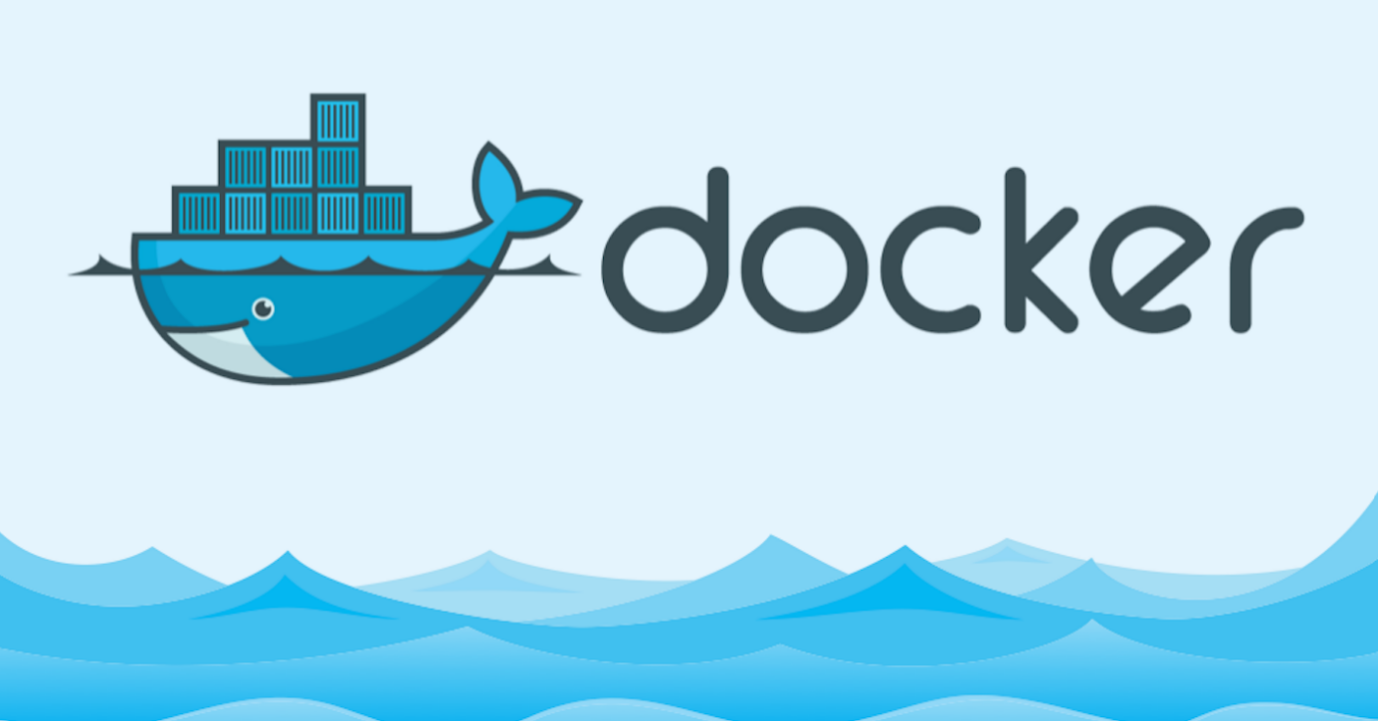


图4-2-6 Docker

## 4.3 技术框架

**4.3.1 Anaconda**

Anaconda是一个开源的Python和R编程环境，旨在简化大规模数据科学和机器学习项目的管理和部署。它包括一个高性能的科学包管理器（Conda），以及一个集成的开发环境（Anaconda Navigator）。



图4-3-1 包管理器Anaconda

1. **科学包管理：** Anaconda附带了Conda包管理器，它可以帮助用户安装、更新和管理Python和R的科学包。Conda还允许您创建独立的虚拟环境，以隔离不同项目的依赖关系，这有助于避免包冲突问题。
2. **预编译二进制包：** Anaconda提供了一个巨大的预编译二进制包库，其中包含了众多用于数据科学、机器学习和科学计算的软件包。这些二进制包可以显著减少安装时间和依赖关系的处理。
3. **跨平台支持：** Anaconda可在多个操作系统上运行，包括Windows、macOS和Linux。这使得用户可以在不同平台上轻松开发和部署数据科学项目。
4. **集成开发环境：** Anaconda Navigator是一个用户友好的图形用户界面（GUI），用于管理环境、安装包和启动应用程序。它包括了Jupyter Notebook、Spyder、RStudio等流行的数据科学工具，使得数据科学家可以方便地进行分析和编程工作。
5. **数据科学库的默认集成：** Anaconda默认包含了一些最常用的数据科学库，如NumPy、Pandas、Matplotlib、SciPy等，这些库构成了数据分析和机器学习的基础。
6. **强大的社区支持：** Anaconda拥有活跃的社区，提供了大量的文档、教程和支持资源，帮助用户解决问题和学习如何使用Anaconda进行数据科学工作。

Anaconda是一个强大的数据科学工具，旨在使数据科学家和机器学习工程师更轻松地管理和部署复杂的项目。它的包管理、虚拟环境和集成开发环境等功能，使得数据科学工作变得更加方便和高效。

**4.3.2 Rasa 框架**

Rasa是一个开源的自然语言处理（NLP）框架，专注于构建和部署任务型对话系统（Task-Oriented Chatbots）和智能助手。它提供了一套工具和库，使开发者能够轻松地创建自定义的、多轮对话的NLP应用程序。



图4-3-2 RASA

1. **任务型对话：** Rasa专注于任务型对话，这意味着它适用于处理用户特定的任务和请求，如查询信息、预订服务、解决问题等。与闲聊型对话不同，任务型对话通常有明确定义的目标。
2. **多轮对话：** Rasa允许开发者构建多轮对话，这意味着对话可以在多个交互轮次中进行，系统可以理解和记住用户在之前轮次中提到的信息，以更自然地进行交流。
3. **自然语言理解（NLU）：** Rasa包含了自然语言理解模块，用于识别用户的意图（Intent）和提取实体（Entities）。它可以处理多语言文本，并支持预训练的NLU模型，如DIETClassifier。
4. **对话管理（Dialogue Management）：** Rasa使用基于规则和机器学习的方法来管理对话流程。开发者可以定义对话策略，以根据用户的意图和实体来决定系统的响应。
5. **自定义操作：** Rasa允许开发者定义自定义操作和动作，以执行各种任务，如查询数据库、调用API、发送消息等。这使得Rasa非常灵活，可以适应不同的应用场景。
6. **集成渠道：** Rasa可以集成到多种渠道中，包括网站、移动应用、Slack、Facebook Messenger等。这使得您可以在各种平台上部署和使用Rasa构建的对话系统。
7. **开源和社区支持：** Rasa是开源的，拥有活跃的社区。这意味着您可以免费使用和修改Rasa，同时可以获得来自社区的支持和贡献。
8. **自定义性：** Rasa提供了广泛的自定义选项，使开发者能够根据项目需求创建高度个性化的对话系统。无论是NLU模型、对话策略还是操作，都可以进行自定义。

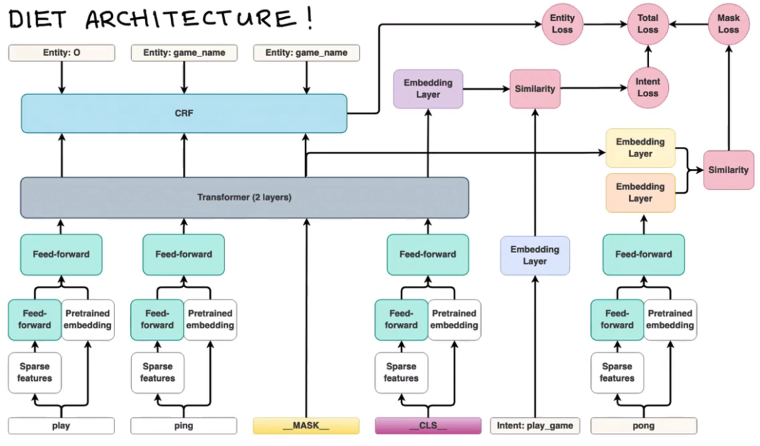
Rasa框架是一个强大的工具，用于构建自定义的任务型对话系统和聊天机器人。它的多轮对话、NLU、对话管理和自定义操作等功能使得开发者可以构建智能的对话应用程序，适用于多种用途，从客户支持到预订服务等。 Rasa的灵活性和开源性质使其成为构建自然语言处理应用的优秀选择。

## 4.4 算法模型

**4.4.1 DIET模型**

Rasa的DIETClassifier（Dual Intent and Entity Transformer Classifier）是自然语言理解（NLU）管道中的关键组件，用于执行多任务文本分类，包括意图识别（Intent Classification）和实体识别（Entity Recognition）。DIETClassifier采用了Transformer架构，是Rasa自然语言处理框架中的重要组成部分。

1. **双重任务：** DIETClassifier能够同时执行意图识别和实体识别两个任务。这意味着它可以在单个组件中处理用户输入，识别用户的意图以及提取任何存在的命名实体。
2. **Transformer架构：** DIETClassifier基于Transformer模型，这是一种深度神经网络架构，以自注意力机制（self-attention）为特色，适用于处理序列数据。Transformer架构已经在自然语言处理任务中取得了显著的成功，如机器翻译和文本分类。
3. **文本编码：** DIETClassifier将用户输入文本编码为固定维度的向量表示，这些向量表示包含了输入文本的语义信息。这些编码的向量可以用于后续的意图识别和实体识别任务。
4. **跨通道联合优化：** DIETClassifier采用了一种跨通道（cross-channel）的联合优化方法，将意图和实体两个任务一起考虑。这有助于提高模型的性能，尤其是在多轮对话中，可以更好地理解用户的意图和提取相关实体。
5. **CRF层：** DIETClassifier还包括一个条件随机场（CRF）层，用于在实体识别任务中建模实体之间的关系和约束。CRF层有助于提高实体标记的准确性。
6. **模型微调：** Rasa的DIETClassifier可以通过微调来适应特定领域或任务的需求。这使得它可以在各种自定义NLU任务中表现出色。

  
图4-4-1 DIET 架构图解

**4.4.2 CRF模型**

条件随机场（Conditional Random Field，CRF）是一种概率图模型，用于建模序列数据的条件概率分布。它常被用于序列标注任务，如命名实体识别（Named Entity Recognition）和词性标注（Part-of-Speech Tagging），以及其他自然语言处理（NLP）任务。CRF广泛用于NLP领域，因为它能够有效地捕捉序列数据中的依赖关系。

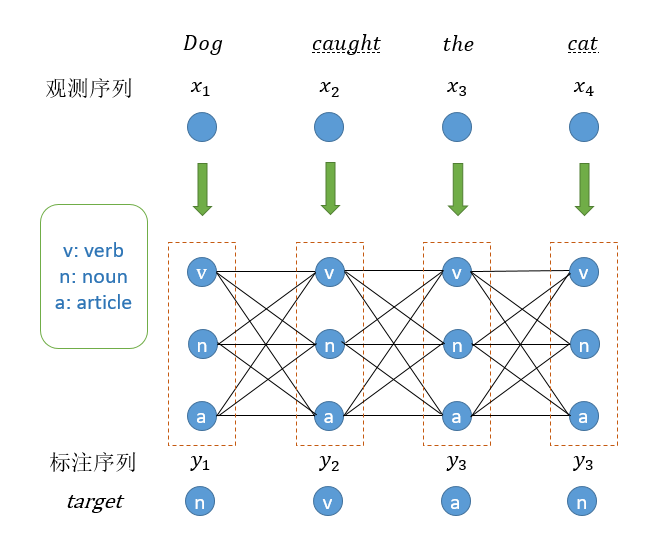


图4-4-2 条件随机场（Conditional Random Field，CRF）

1. **序列建模：** CRF被设计用来对序列数据进行建模，其中包括一个输入序列和一个输出序列。在NLP中，输入序列通常是一句话或文本段落，而输出序列是相应的标签序列，用于标注输入序列中的各个部分。
2. **条件概率分布：** CRF建模的是给定输入序列条件下输出序列的条件概率分布。这表示CRF考虑了输入序列的影响，以便更准确地进行标注。
3. **特征函数：** CRF的条件概率分布由一组特征函数组成，这些函数对输入序列和输出标签序列的组合进行评分。每个特征函数都有一个相关的权重，通过学习得到。特征函数可以捕捉到各种不同的依赖关系，如词汇特征、上下文信息等。
4. **全局归一化：** CRF使用全局归一化因子来确保生成的标签序列的概率总和为1。这有助于解决标签之间的依赖性问题，确保生成的标签序列是合理的。
5. **解码：** CRF使用不同的解码算法来确定最佳的输出标签序列，以最大化条件概率分布。常见的解码算法包括维特比算法（Viterbi Algorithm）和信念传播算法（Belief Propagation）。

在自然语言处理任务中，CRF常用于命名实体识别（NER）等序列标注任务，其中需要识别文本中的实体（如人名、地名、日期等）。CRF的能力在处理上下文依赖性和序列结构方面非常强大，因此在NLP中得到了广泛的应用。

在Rasa框架中，CRF常用于实体识别任务，帮助识别对话中的命名实体，如日期、地点、人名等。CRF层被添加到DIETClassifier中，以提高实体识别的性能和准确性。这使得Rasa在构建任务型对话系统时能够更好地理解用户提供的信息。

# 第五章 项目研究过程

## 5.1 项目研究方法

研究方法，是指在研究中发现新现象、新事物，或提出新理论、新观点，揭示事物内在规律的工具和手段。这是运用智慧进行科学思维的技巧，一般包括[文献调查法](https://baike.baidu.com/item/%E6%96%87%E7%8C%AE%E8%B0%83%E6%9F%A5%E6%B3%95/5141100)、[观察法](https://baike.baidu.com/item/%E8%A7%82%E5%AF%9F%E6%B3%95/1210099)、思辨法、[行为研究法](https://baike.baidu.com/item/%E8%A1%8C%E4%B8%BA%E7%A0%94%E7%A9%B6%E6%B3%95/12615489)、历史研究法、[概念分析法](https://baike.baidu.com/item/%E6%A6%82%E5%BF%B5%E5%88%86%E6%9E%90%E6%B3%95/7250155)、[比较研究法](https://baike.baidu.com/item/%E6%AF%94%E8%BE%83%E7%A0%94%E7%A9%B6%E6%B3%95/5566125)等。研究方法是人们在从事科学研究过程中不断总结、提炼出来的。由于人们认识问题的角度、研究对象的复杂性等因素，而且研究方法本身处于一个在不断地相互影响、相互结合、相互转化的动态发展过程中，所以对于研究方法的分类很难有一个完全统一的[认识](https://baike.baidu.com/item/%E8%AE%A4%E8%AF%86/8581906)。

①文献研究法

[文献研究法](https://baike.baidu.com/item/%E6%96%87%E7%8C%AE%E7%A0%94%E7%A9%B6%E6%B3%95/3668258)是根据一定的研究目的或课题，通过调查文献来获得资料，从而全面地、正确地了解掌握所要研究问题的一种方法。文献研究法被广泛用于各种学科研究中。其作用有：①能了解有关问题的历史和现状，帮助确定研究课题。②能形成关于研究对象的一般印象，有助于观察和访问。③能得到现实资料的比较资料。④有助于了解事物的全貌。

②实证研究法

实证研究法是科学实践研究的一种特殊形式。其依据现有的科学理论和实践的需要，提出设计，利用科学仪器和设备，在自然条件下，通过有目的有步骤地操纵，根据观察、记录、测定与此相伴随的现象的变化来确定条件与现象之间的因果关系的活动。主要目的在于说明各种自变量与某一个因变量的关系

③数学建模法

数学方法就是在撇开研究对象的其他一切特性的情况下，用[数学工具](https://baike.baidu.com/item/%E6%95%B0%E5%AD%A6%E5%B7%A5%E5%85%B7)对研究对象进行一系列量的处理，从而作出正确的说明和判断，得到以数字形式表述的成果。科学研究的对象是质和量的统一体，它们的质和量是紧密联系,质变和量变是互相制约的。要达到真正的科学认识，不仅要研究质的规定性，还必须重视对它们的量进行考察和分析，以便更准确地认识研究对象的本质特性。数学方法主要有统计处理和[模糊数学](https://baike.baidu.com/item/%E6%A8%A1%E7%B3%8A%E6%95%B0%E5%AD%A6)分析方法。

## 5.2 项目环境配置

在项目的开始时，我们首先需要配置好项目所需的运行环境，安装相应的依赖包。所需安装的包名及版本如下：

|  |  |
| --- | --- |
| Package | Version |
| rasa | 3.1 |
| pymysql | 1.1.0 |
| py2neo | 2021.2.3 |
| pandas | 1.5.3 |
| numpy | 1.23.5 |
| scipy | 1.8.1 |
| matplotlib | 3.5.3 |
| axios | 1.4.1 |
| d3 | 7.8.5 |
| element-ui | 2.15.13 |
| babel-plugin-component | 1.1.1 |
| highcharts | 11.1.0 |
| less | 4 |
| less-loader | 7 |
| vue-router | 3 |
| vuex | 3 |
| flask | 1.1.2 |

使用docker构建部署项目，镜像使用清单：

|  |  |  |
| --- | --- | --- |
| image | baseImage | Package |
| philoboy/rasa\_zh\_md:1.0 | rasa/rasa:3.5.1 | spacy zh\_core\_web\_md |
| philoboy/rasa-sdk:1.6 | rasa/rasa-sdk:3.5.1 | py2neo 2021.2.3  pandas 1.5.3  matplotlib 3.5.3  numpy 1.23.5  pyecharts 2.0.4  pymysql 1.1.0rc2  scipy 1.11.2  requests 2.31.0 |

系统运行环境配置如下：

|  |  |
| --- | --- |
| CPU：14核  内存：32GB | 13th Gen Intel(R) Core(TM) i5-13500HX |
| GPU | NVIDIA GeForce RTX 4060 Laptop GPU 8GB |
| 数据盘：1TB SSD  最大扩容：3TB |  |
| 操作系统： | Windows 11 专业版 |
| 软件环境:部分依赖包 | conda 23.3.1  python 3.10  rasa 3.1  pymysql 1.1.0  py2neo 2021.2.3  pandas 1.5.3  numpy 1.23.5  scipy 1.8.1  matplotlib 3.5.3  axios 1.4.0  d3 7.8.5  element-ui 2.15.13  highcharts 11.1.0  less 4  less-loader 7  vue-router 3  vuex 3  flask 1.1.2 |

## 5.3 项目数据集

本项目所使用的全部河道站信息均来自全国水雨情信息网

全国水雨情信息数据集：

全国水雨情信息网是中国国家气象局管理的一个重要平台，旨在监测和发布全国范围内的降雨情况和水文信息。

1. **监测和数据收集：** 全国水雨情信息网主要用于监测和收集全国范围内的降雨数据和水文信息。这些数据包括降雨强度、降水时空分布、河流水位、水流速度、水质等。
2. **实时更新：** 该网站提供实时更新的数据，使政府、气象部门、水利部门和公众能够及时了解当前的降雨情况和水文情况。
3. **应急响应：** 全国水雨情信息网在自然灾害和突发事件发生时发挥着关键作用。例如，在洪水、暴雨或其他降雨引发的紧急情况下，该网站的数据和信息可以用于预警、应急响应和灾害管理。
4. **政府决策支持：** 政府部门和决策者可以利用全国水雨情信息网的数据来制定政策、规划和管理水资源、防灾减灾等方面的工作。
5. **公众信息传播：** 该网站也向公众提供降雨和水文信息，以帮助居民更好地理解和应对不同气象条件下的风险。



图5-3-1 全国水雨情信息网

# 第六章 项目研究结果

## 6.1 意图实体识别优化结果

在Rasa中，意图（intent）和实体（entity）的识别是构建自然语言处理（NLP）聊天机器人的关键组成部分。

意图的识别率在于nlu中意图举例数量，而实体能否准确识别并装入槽(slot)确是一个麻烦的问题。slot有很多种类型,from\_entity类型是从实体装槽(slot),from\_text则是无论用户输入的是什么，都会将其装入槽中。最常用的就是from\_entity类型。只需在nlu中intent举例时将实体标识出来。但由于多义词的存在，有的时候虽然出现了这个实体的名称，但可实际上我表达的可能是这个词的另一个意思，我并不期望slot去装这个实体，但是rasa并不能理解我们对话的意思，它只是按照预设地出现某个词就将其捕获放入槽中。那么就会影响对话，出现unexpected的情况。

使用Rasa form有助于机器人系统在对话中了解用户的意图和需求。，在我们的训练数据中大量使用了form优化意图和需求识别。

"form"用于指导对话中的槽值收集过程。通过定义一个表单（form），可以指定哪些槽值需要在对话中收集，以及它们的类型。如果用户没有提供所需的信息，"form"可以提示用户提供缺失的信息，以便完成槽值的收集。这有助于改善对话的交互流程。"form"还支持中断处理，这意味着在槽值收集过程中，机器人可以处理一些紧急的用户请求或其他任务，然后回到收集槽值的流程，而不会丢失上下文。

# 第七章 项目分析与展示

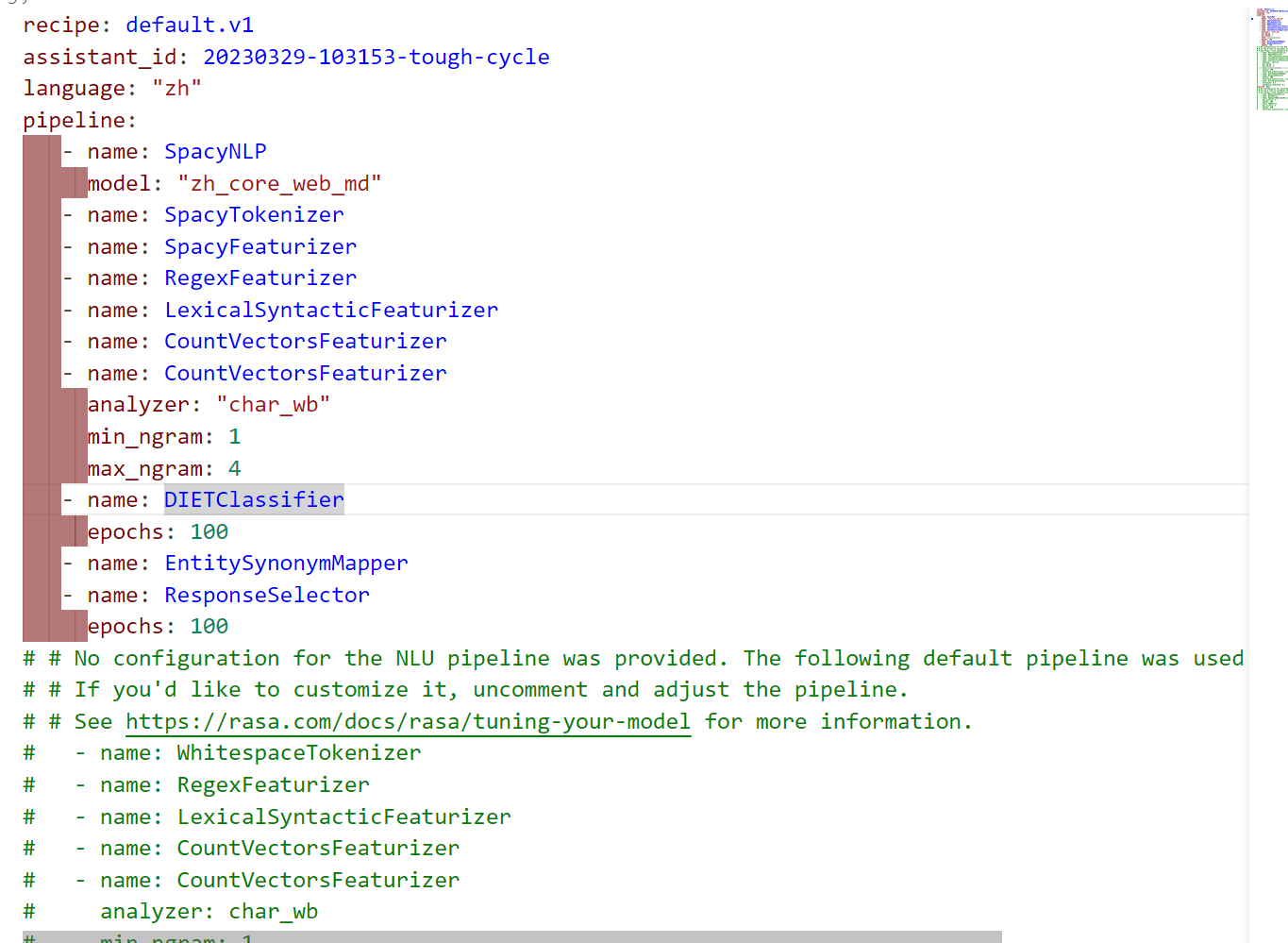
## 7.1 技术难点

**7.1.1 中文对话实体意图识别技术难点**

1、Rasa默认并不支持中文，且其预载的管道(pipline)和语言支持也没有中文，只有英语法语之类的以空格分词的语言。中文相对英文分词更加困难，pipline的原理也是十分复杂

在项目早期，我们就注意到了Rasa对中文并不支持。 Rasa管道(pipline)是一个由一系列NLP（自然语言处理）组件组成的配置，用于处理和分析用户输入文本以及生成对话系统的响应。管道决定了Rasa如何解析、理解和生成对话。所以为了支持中文，我们就要设置针对中文的管道。

好在rasa支持导入与预训练模型，我们选定了Spacy的中文训练包。经过对管道的研究，我们设定好了pipline



SpacyNLP：使用Spacy自然语言处理库进行文本处理，model参数指定了使用的Spacy模型，这里是"zh\_core\_web\_md"，表示中文的模型。

SpacyTokenizer：使用Spacy进行分词，将文本拆分成单词或标记。

SpacyFeaturizer：使用Spacy提取文本特征，以获取更丰富的语义信息。

RegexFeaturizer：使用正则表达式提取文本特征，用于捕获特定的模式。

LexicalSyntacticFeaturizer：提取词汇和句法特征。

CountVectorsFeaturizer：使用词袋模型提取文本特征，用于将单词转换为向量表示。此组件在配置中出现两次，一个用于词级别分析，另一个用于字符级别分析（指定了analyzer: "char\_wb"，以及min\_ngram和max\_ngram参数来控制字符级别的n-gram范围）。

DIETClassifier：这是一个深度学习模型，用于同时进行意图分类和实体提取。epochs参数指定了训练的轮数。

EntitySynonymMapper：用于处理实体的同义词映射，以确保不同的表达方式可以映射到同一个实体。

ResponseSelector：用于从事先定义的响应中选择最合适的回复，例如用于聊天中的自动回复。epochs参数指定了训练的轮数。

2、编写简单数据意图识别混乱，抓取到关键字就填槽(slot)，导致对话混乱，功能紊乱。

在Rasa中，实体（Entity）是指用户消息中具有特定含义的文本片段或信息。实体通常是对话中需要被识别、提取和理解的重要部分，例如日期、时间、地点、人名、产品名称等等。

要实现诸如降水量查询和洪水传播时间查询的功能，需要将用户消息中的地点实体提取出来。

我们使用槽(slot)来实现存储这些关键实体。槽（Slot）是一个用于存储和跟踪对话状态信息的关键概念。槽通常用于存储对话中的关键数据，如用户提供的信息、对话的上下文、实体识别的结果等等。

使用实体(Entity)和槽(slot)便能实现一系列功能，比如降水量查询，在训练数据中使用[]标记实体并说明实体类型。然后定义好存放这个实体类型的slot。使用slot去数据库查询。slot就相当于一个变量一样。

但如果仅仅是这样简单地使用slot，有很多缺陷。如果用户按照程序员想的那样按步骤输入数据，称为happy path，但是实际会出现很多unhappy path。

比如我们做一个让rasa记住我们的T-shirt 大小的功能，当我们告诉rasa我的T-shirt大小后，rasa应该记住它。所以我们要定义了一个shirt\_size的slot来存储它，但假如发生如下对话

me: hi

rasa: hello,there. How are you?

me: medium. I feel sad.

rasa: Thanks, I'll remember that.

me: repeat my shirt size

rasa: Your shirt size is medium!

我们在回复rasa问我们感觉怎么样时回复了medium，虽然我们是在说自己的心情，但是rasa并不知道这个。它会以为这个medium是T-shirt size。那么虽然我们之前我们没有告诉rasa我的shirt size，但是它就会回复它是medium。这显然不是我们想要的，可以想象到如果这么去写，那么当我们的功能多了起来，那么对话明显会紊乱。因为slot的填入与否也会影响rasa的意图预测，继而影响对话。

slot并不够结构化，它只是零散地记录出现在对话中的实体，并且因为一词多义，rasa会错误地填槽。

理解缺陷后，我们选择在domain中加入form section

Rasa Forms（表单）是Rasa框架中的一种高级功能，用于处理用户输入并管理对话中的多轮交互。表单允许您以一种结构化的方式收集用户信息，并确保在填写表单期间用户提供的信息完整、有效。

1. slots:
2. shirt\_size:
3. type: text
4. influence\_conversation: true
5. mappings:
6. - type: from\_entity
7. entity:shirt\_size
8. conditions:
9. - active\_loop: shirt\_size\_form
10. forms:
11. shirt\_size\_form:
12. required\_slots:
13. - shirt\_size

在rule中加入触发form的规则，当满足条件后触发form(trigger form)。因为slot中我们设定只有form被触发后才会填这个槽(slot)，因此就阻止了上面提到的现象的发生。

1. - rule: Activate form
2. steps:
3. - intent: ask\_me\_anything
4. - action: shirt\_size\_form
5. - active\_loop: shirt\_size\_form
7. - rule: Submit form
8. condition:
9. - active\_loop: shirt\_size\_form
10. steps:
11. - action: shirt\_size\_form
12. - active\_loop: null
13. - slot\_was\_set:
14. - requested\_slot: null
15. - action: utter\_remember

此外还会出现很多unhappy path，都是本个项目的难点，也都可以通过forms和恰当的rules解决

**7.1.2 自定义动作技术难点**

Rasa 框架允许使用自定义动作来扩展对话机器人的功能。自定义动作是在actions.py文件中编写的代码，用于执行各种任务，如与外部系统交互、查询数据库、执行计算等，可以说是Rasa框架的核心内容之一。

在Rasa中创建自定义动作有着固定的步骤：

1. **创建自定义动作脚本：** 需要创建一个 Python 脚本来定义您的自定义动作。Rasa项目在初始化时会生成一个actions.py文件，在这里可以编写自定义脚本。
2. **导入必要的库和模块：** 在自定义动作脚本中，通常需要导入 Rasa SDK 和其他必要的 Python 库或模块，以便与 Rasa 框架进行交互。
3. **from** rasa\_sdk **import** Action, Tracker
4. **from** rasa\_sdk.executor **import** CollectingDispatcher
5. **创建自定义动作类：** 在自定义动作脚本中，需要创建一个继承自 Action 类的自定义动作类，并实现 run 方法。run 方法包含了自定义动作的代码逻辑。一个自定义动作类遵循以下格式：
6. **class** YourCustomAction(Action):
7. **def** name(self):
8. # 指定自定义动作的名称，用于在对话中引用
9. **return** "your\_custom\_action\_name"
11. **def** run(self, dispatcher, tracker, domain):
12. # 自定义动作的逻辑代码
14. **return** []

在项目开发过程中，在自定义动作编写这部分我们也遇到了许多的问题：

1. 在编写自定义动作时获取用户对话中的实体值

这个问题可以通过使用rasa\_sdk中提供的tracker中的getslot方法来取得，例如在ActionSearchPrecipitationByName这个自定义类中，使用getslot来获得用户对话中提供的place实体值：

1. **class** ActionSearchPrecipitationByName(Action):
2. **def** name(self) -> Text:
3. **return** "action\_search\_precipitation\_by\_name"
4. **def** run(self, dispatcher: CollectingDispatcher,
5. tracker: Tracker,
6. domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
8. place=tracker.get\_slot("place")

在getslot方法中传入对于槽的名字则可获取对应槽的值。

1. 从数据库中获取河道站相关数据

在本项目中解决这个问题时通过导入pymysql这个包来实现的，pymysql 是 Python 中一个用于连接和操作 MySQL 数据库的开源库。它通过 Python 代码执行各种数据库操作，包括查询、插入、更新和删除数据，以及管理数据库连接。在ActionSearchPrecipitationByName这个自定义类中，我们通过使用pymysql来从数据库中获取河道站数据，依此执行查询操作：

1. # 连接数据库
2. conn = pymysql.connect(host='43.142.246.112', port=3306, user='common', password='common666', db='hydrology', charset='utf8')
3. cur = conn.cursor(pymysql.cursors.DictCursor) # 生成游标对象
4. sql=f"select \* from hydrometric\_station where name='{place}'"
5. cur.execute(sql)
6. data=cur.fetchall()

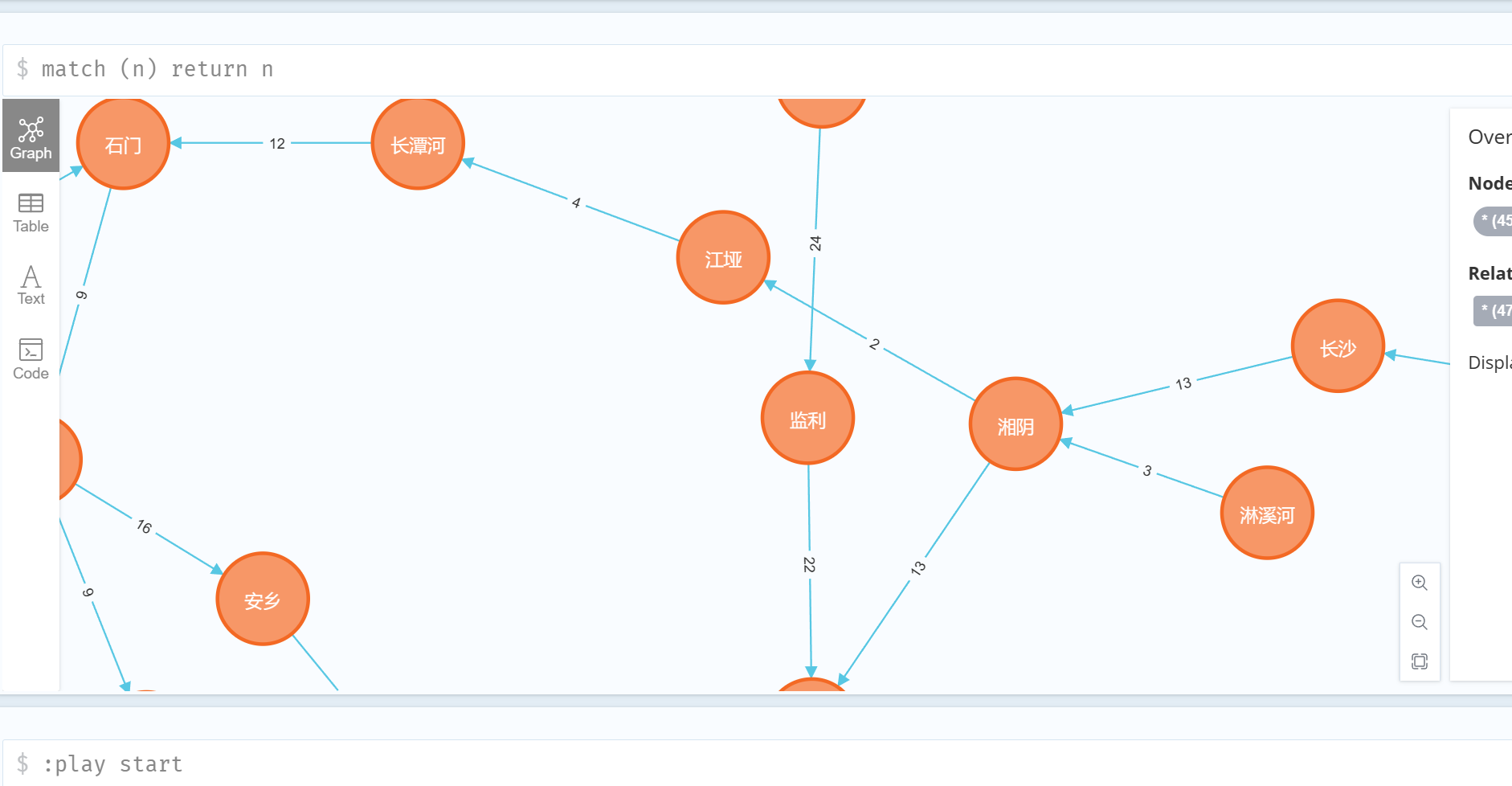
3、消除以填充槽对后续对话的影响

在Rasa框架中一个槽如果已经被rasa捕捉并填充，则会对模型对后续意图的预测概率产生影响，所以这就需求我们在使用完槽后如果确认后续不需要使用该槽或者该槽需要被赋予新的值，则需要将槽释放，在本项目中选取的释放槽方案为，当对话调用完一个自定义动作后，使用return将槽中的值重置为None：

1. resetSlot=None # 将槽重置
3. **return**[SlotSet('place', resetSlot)]

在这里我们将resetSlot这个变量的值赋值为None再使用SlotSet方法将resetSlot的值赋值给槽，槽值重置为None则表明槽已被清空可以获取新的值。

4、从图数据库获取洪水传播数据进行处理并使用Dijkstra算法计算出洪水传播时间



这是我们的图数据库，要计算两地洪水传播时间，就是算图上这两点的最短路径，那么就要使用Dijkstra算法。Dijkstra算法的代码实现涉及到邻接矩阵，只要能拿到邻接矩阵(二维list)，就能算出一个点到其余点的最短路径。

数据处理是一大难点。

1. graph = Graph('bolt://43.142.246.112:7687', auth=('neo4j', 'common666'))
2. cypher1 = "match (n:riverstation) return n.name as node"
3. cypher2 = ("match (n:riverstation)-[r]->(m:riverstation) return STARTNODE(r) as source,ENDNODE(r) as target,"
4. "r.time as time")
5. node\_df = graph.run(cypher1).to\_data\_frame()
6. edge\_df = graph.run(cypher2).to\_data\_frame()

首先这样连接数据库，并定义查询语句查询并返回转换成DataFrame存在变量中便于之后处理

然后因为node\_df知识查询站点并返回名字得到的，所以直接将node\_df的name列转换成list，那么这个list的每个元素就是str类型的。

但是edge\_df接受的是关系，每一个元素的数据类型是py2neo里的Node类型，所以需要再将里面的站点名转换成str类型

1. node\_list = node\_df["node"].tolist()
2. source\_list = edge\_df["source"].tolist()
3. target\_list = edge\_df["target"].tolist()
4. time\_list = edge\_df["time"].tolist()
5. edge\_list = list()
6. **for** i **in** range(len(source\_list)):
7. edge\_list.append([source\_list[i]["name"], target\_list[i]["name"], time\_list[i]])e()

转换为str后就放入edge\_list中，然后就构造邻接矩阵

1. inf = 999999
2. adj\_matrix = [[inf **for** i **in** range(len(node\_list))] **for** j **in** range(len(node\_list))]
3. **for** i **in** range(len(adj\_matrix)):
4. adj\_matrix[i][i] = 0
5. **for** relation **in** edge\_list:
6. x = node\_list.index(relation[0])
7. y = node\_list.index(relation[1])
8. adj\_matrix[x][y] = relation[2]

最后的Dijkstra也是难点，要理解并写出代码也是有难度

1. start = node\_list.index(source\_station)
2. end = node\_list.index(destination\_station)
3. # dijkstra算法
4. passed = [start]
5. nopass = [x **for** x **in** range(len(adj\_matrix)) **if** x != start]
6. dis = adj\_matrix[start]
8. **while** len(nopass):
9. idx = nopass[0]
10. **for** i **in** nopass:
11. **if** dis[i] < dis[idx]: idx = i
12. nopass.remove(idx)
13. passed.append(idx)
14. **for** i **in** nopass:
15. **if** dis[idx] + adj\_matrix[idx][i] < dis[i]: dis[i] = dis[idx] + adj\_matrix[idx][i]
16. dispatcher.utter\_message(f"从{source\_station}到{destination\_station}的洪水传播时间为{dis[end]} h")
17. **return** [SlotSet("source\_station", None), SlotSet("destination\_station", None)]

**7.1.3 前端开发技术难点**

1、登录界面和主界面展示的选择

对于SPA，在显示登录界面和主界面的时候如何做出选择这一方面遇到难点，通过查阅资料，发现在vue路由导航的时候可以通过meta来区分。

1. routes:[
2. {
3. path:"/login",
4. name:"login",
5. component:Login,
6. meta:{
7. keepalive:**false**
8. }
9. },
10. {
11. path: "/",
12. name:"main",
13. component: Main,
14. meta:{
15. keepalive: **true**
16. }
17. },
18. ...后续代码省略
19. ]

对于每一次路由访问，只需要通过前置路由守卫判断用户要去往的界面，并对用户是否登录进行校验，就既能区分登录界面和主界面，又能实现安全校验功能。

1. router.beforeEach((to,from,next)=>{
2. **if**(to.meta.keepalive){
3. **if**(localStorage.getItem("login")==="login"){
4. next()
5. }**else**{
6. router.push("/login")
7. }
8. }**else**{
9. next()
10. }
11. })

然后在App.vue中写两个el-container，通过v-if=”$route.meta.keepalive”决定具体显示哪一个界面。

1. <**template**>
2. <div>
3. <el-container **class**="home-container" v-**if**="$route.meta.keepalive">
4. <el-header>
5. <Header></Header>
6. </el-header>
7. <el-container>
8. <el-aside width="200px"><Aside></Aside></el-aside>
9. <el-main><router-view></router-view></el-main>
10. </el-container>
11. </el-container>
12. <router-view v-**if**="!$route.meta.keepalive"></router-view>
13. </div>
14. </**template**>

2、水位流量关系线的获取与展示

对于水位流量关系线的展示，我们选择的是通过后端连接数据库并绘图，然后将绘制的图片传递到前端。

通过flask的send\_file方法将图片传到前端，并设置mimetype=”image/png”

1. @app.route("/get\_water\_flow\_data", methods=["post"])
2. def get\_water\_level\_data():
3. base\_path="./water\_flow\_charts"
4. place=request.json.get("place")
5. img\_path=f"{base\_path}/{place}.png"
7. **if** not os.path.exists(img\_path):
8. get\_water\_flow\_data(place)
10. **return** send\_file(img\_path,mimetype="image/png")

前端接收图片的时候，需要指定responseType:”blob”，以告诉axios将响应数据解析为Blob对象。当成功获取响应后，我们创建了一个Blob对象并使用window.URL.createObjectURL()方法生成一个URL，然后将这个URL赋值给imgUrl。这样，图片就会在<img>标签中显示。

1. **if**(matchedStation){
2. axios.post("/charts/get\_water\_flow\_data",{place:matchedStation},{responseType:"blob"})
3. .then(response=>{
4. **this**.messages.push({text:"水位流量关系线如下：",from:"rasa",prompt:"防洪减灾机器人："})
6. **const** blob = **new** Blob([response.data],{type:"image/png"})
7. **const** url=window.URL.createObjectURL(blob)
8. **this**.imgSrc=url
9. })
10. .**catch**(error=>{
11. console.log("获取水位流量关系线图片失败："+error)
12. })
13. }

显然，最后一个问题就是如何保证水位流量关系线显示并只显示一次，这里可以通过v-if="index === messages.length - 1 && imgSrc" 进行判断。

1. <div v-**if**="index === messages.length - 1 && imgSrc">
2. <img :src="imgSrc" alt="Water Flow Chart">
3. </div>

## 7.2 项目展示

**7.2.1 登录界面展示**

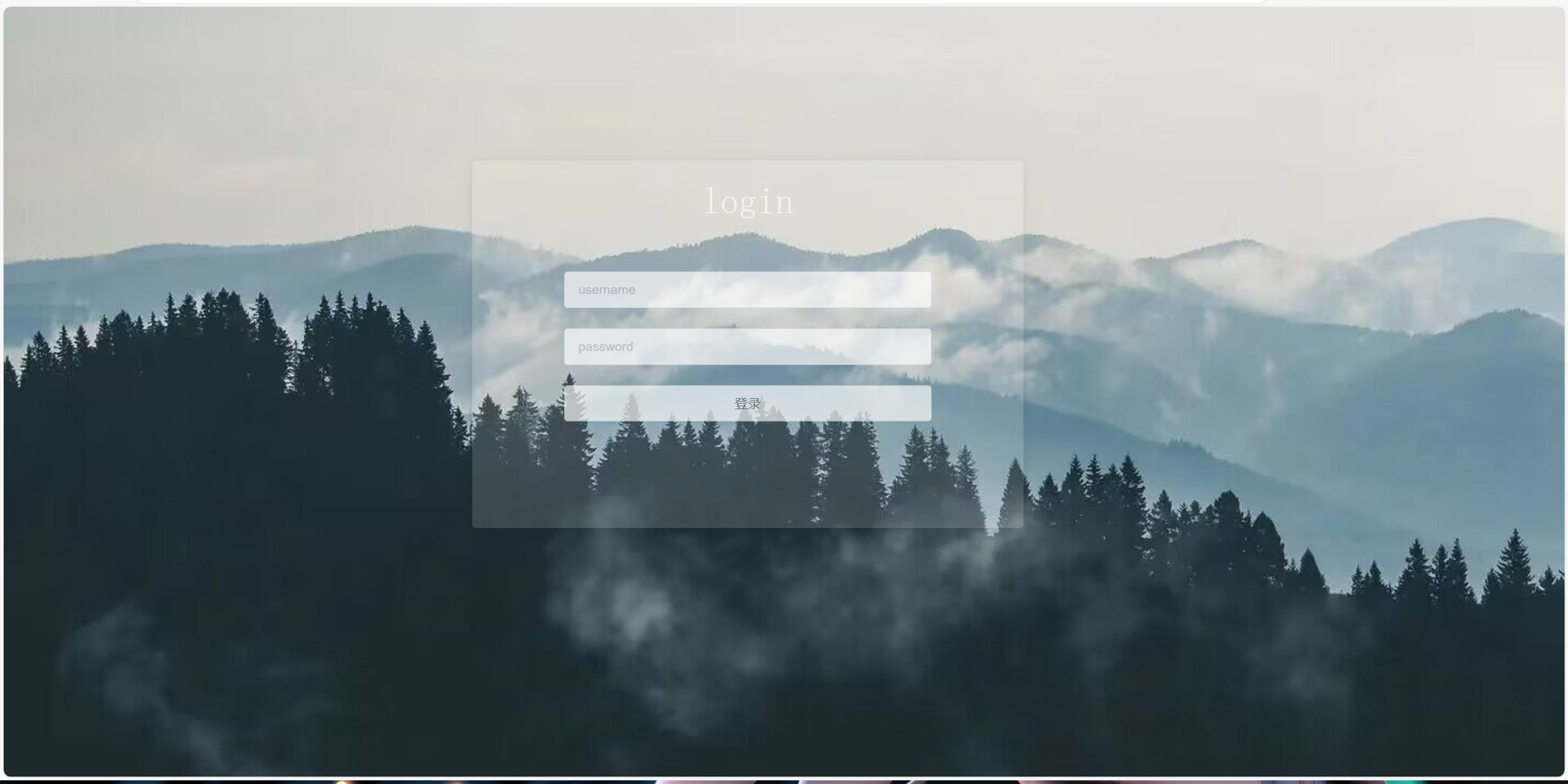
****

图7-2-1 登录界面

这是我们多轮对话防洪减灾机器人的登录界面，默认账号密码均为rasa，正确输入即可进入防洪减灾问答机器人系统。

**7.2.2 系统主界面展示**



图7-2-2 系统主界面

系统主界面主要有两个功能模块分别为**问答专栏**和**知识图谱**

其中**问答专栏**可以完成的功能有人机对话窗口、水位流量关系线查询、洪水传播时间查询、河道站查询、降水查询。

知识图谱模块则主要展示河道站点的知识图谱。

**7.2.3 人机对话功能展示**

接下来我将与机器人完成一些简单的对话：

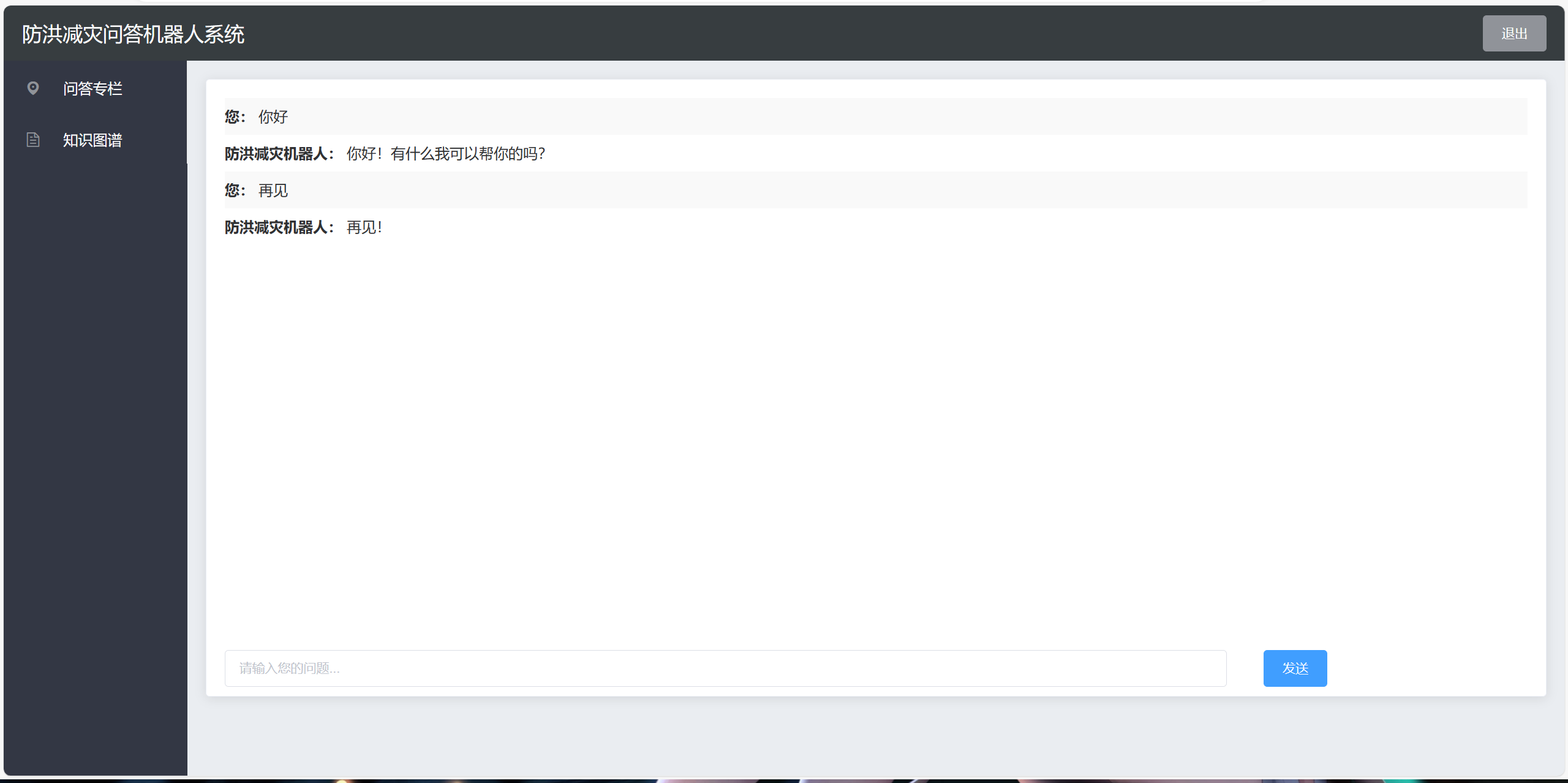


图7-2-3 简单的人机对话

可以看见机器人回答了这些问题。

**7.2.4 河道站查询功能展示**

接下来我将去询问机器人**寸滩**河道站的信息



图7-2-4 河道站查询功能

机器人成功查询到了**寸滩**河道站的信息

**7.2.5 降水查询功能展示**

接下来我将去查询**沙市**的降水量

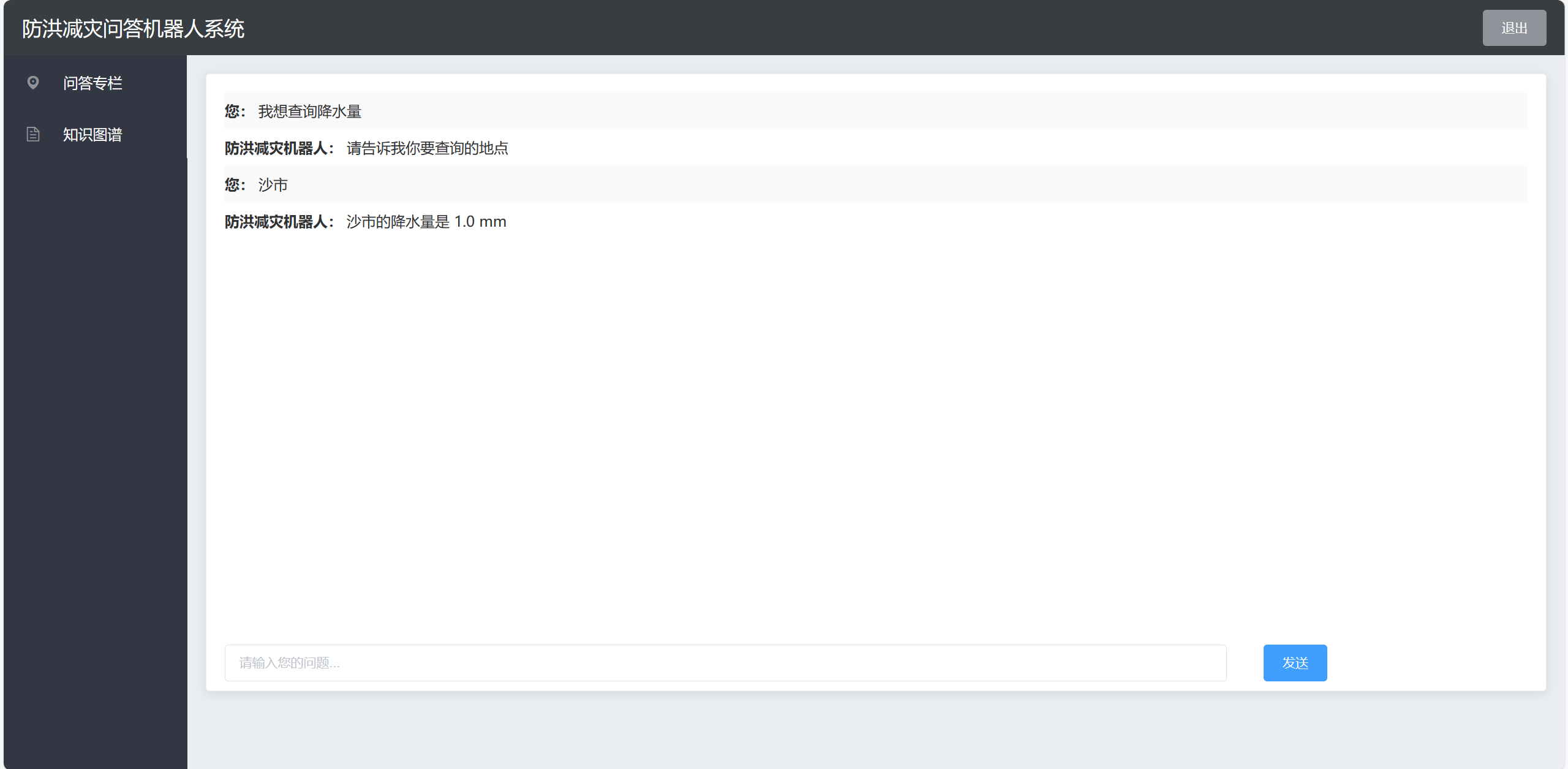


图7-2-5 降水查询功能

机器人成功查询到了**沙市**的降水量

**7.2.6 洪水传播时间查询功能展示**

接下来我将询问机器人**寸滩**到**枝城**的洪水传播时间

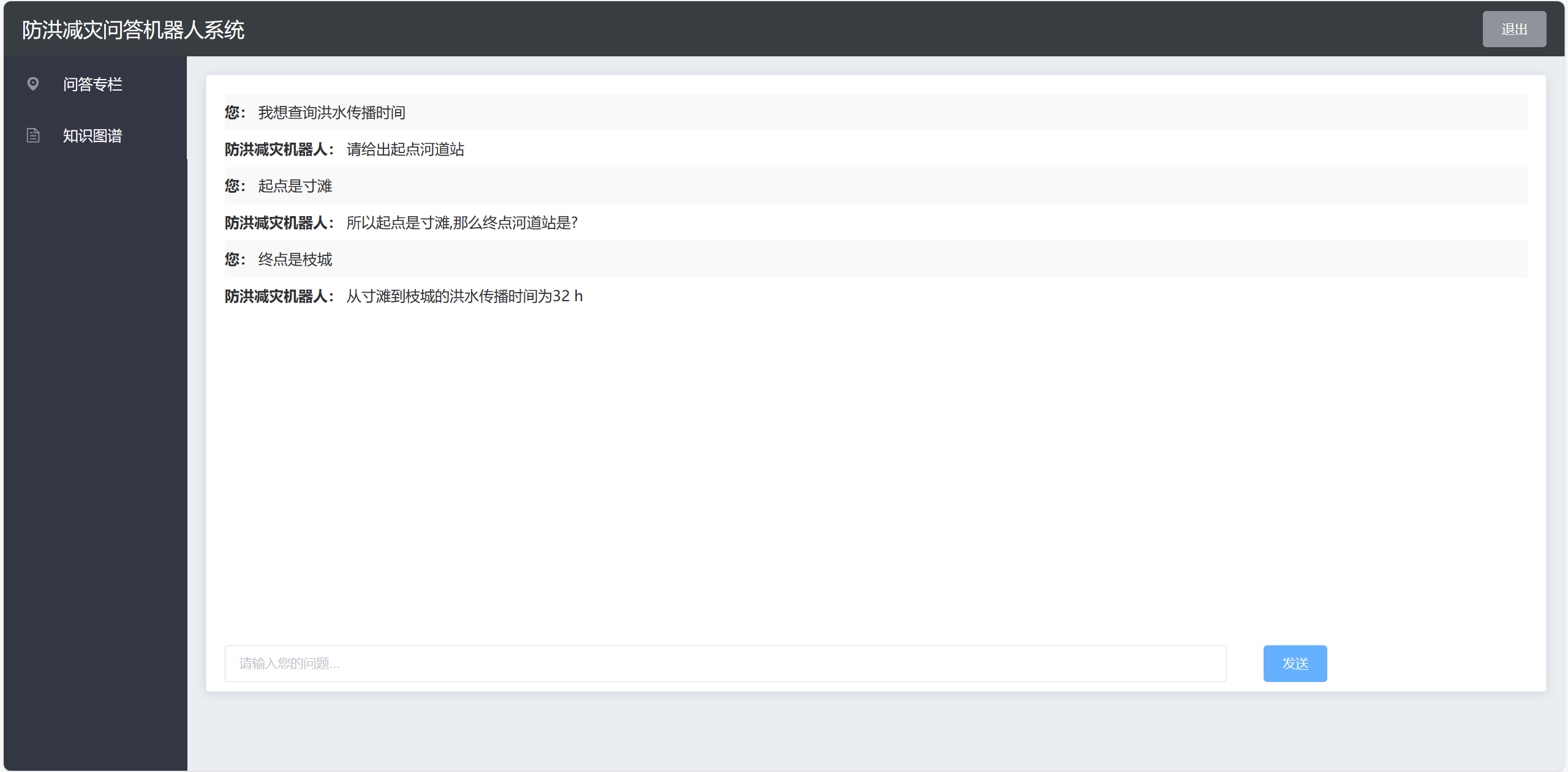


图7-2-6 洪水传播时间查询功能

机器人成功计算出了**寸滩**到**枝城**的洪水传播时间，通过后台数据查询结果正确

**7.2.7水位流量关系线展示**

下面我将询问机器人**桃江**的水位流量关系线



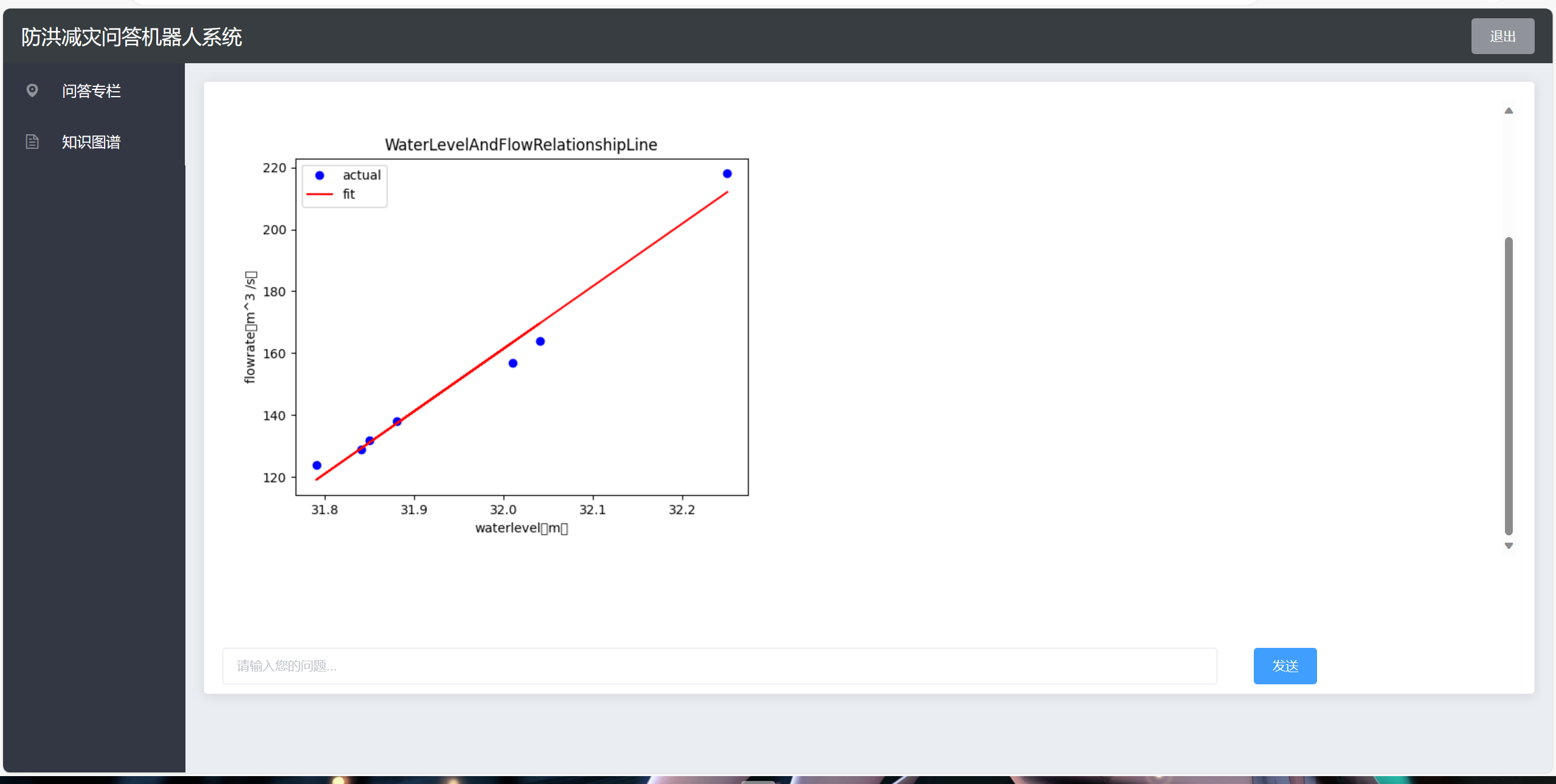
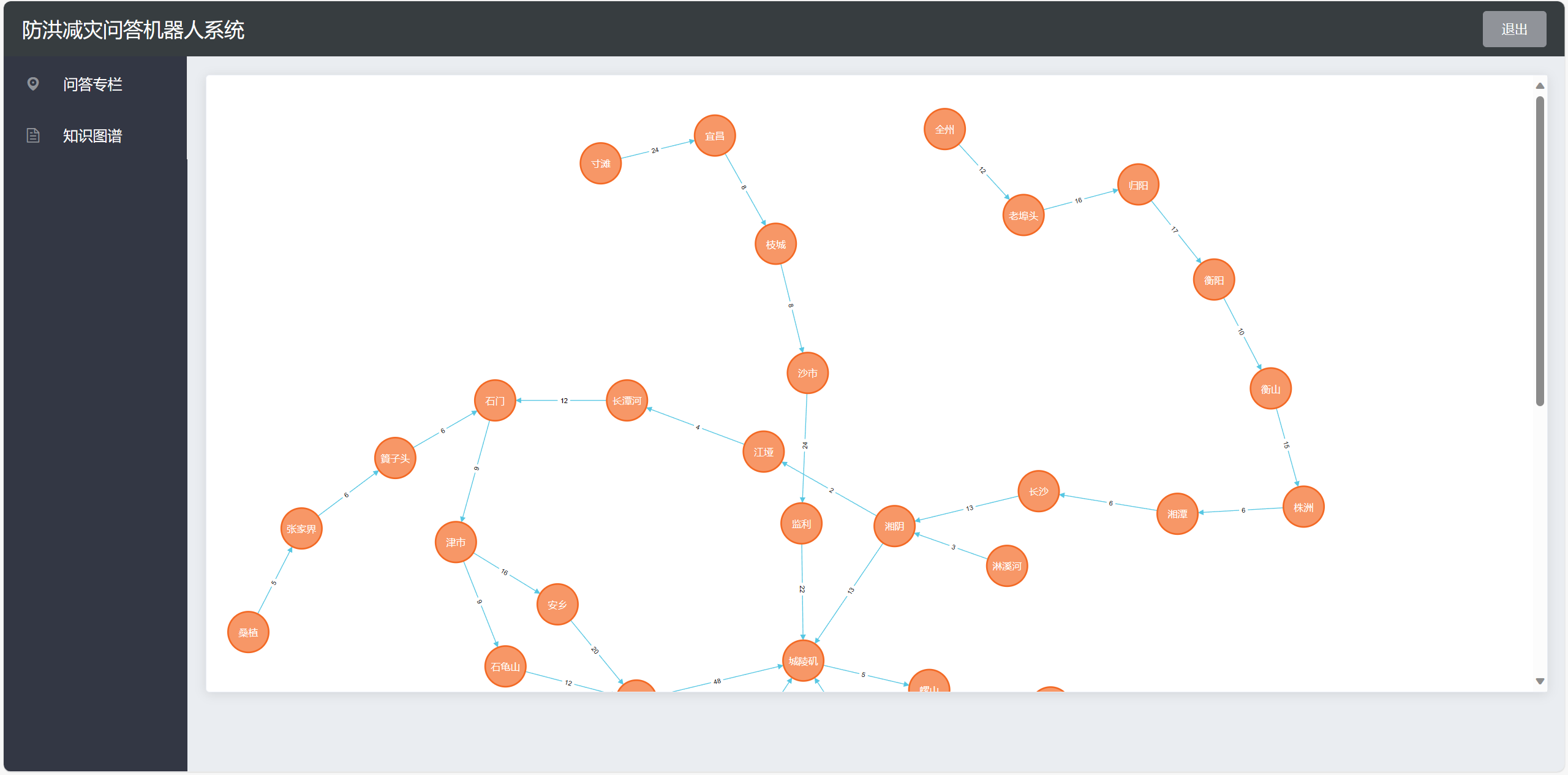


图7-2-7 水位流量关系线功能

机器人成功生成了**桃江**的水位流量关系线

**7.2.8知识图谱展示**

点击知识图谱功能查看



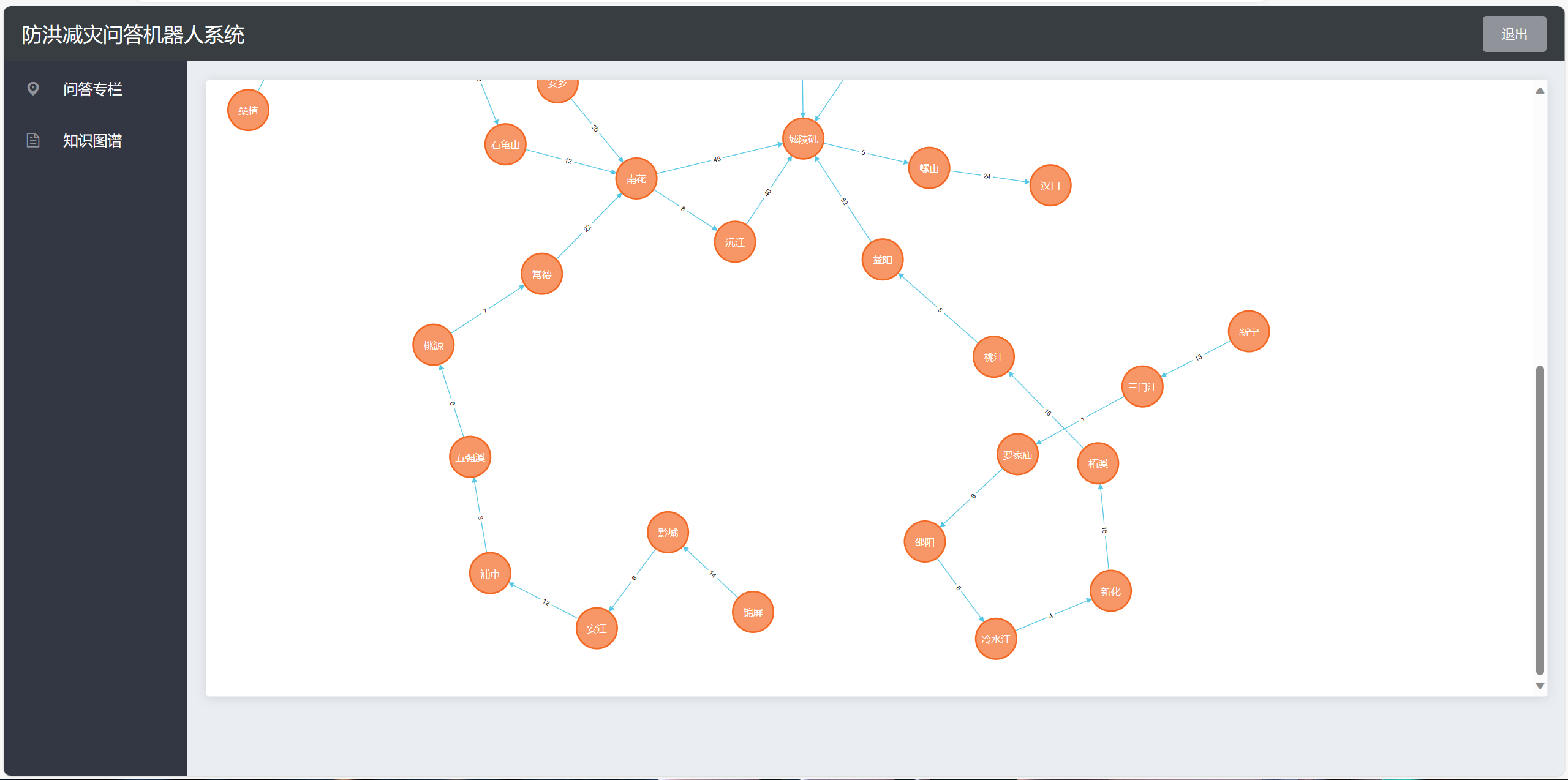


图7-2-8 知识图谱

网页成功展示出了知识图谱

# 第八章 参考资料

1. Rasa Open Source：[Conversational AI Platform | Superior Customer Experiences Start Here (rasa.com)](https://rasa.com/)
2. 全国水雨情信息网：[全国水雨情信息 (mwr.cn)](http://xxfb.mwr.cn/sq_djdh.html)
3. Py2neo Handbook：[The Py2neo Handbook — py2neo 2021.1](https://py2neo.org/2021.1/)
4. Docker Docs：[Get Docker | Docker Docs](https://docs.docker.com/get-docker/)
5. Element：<https://element.eleme.cn/2.13/#/zh-CN/component/installation>
6. Vue：https://cn.vuejs.org/guide/introduction.html