文章编号：2095-2163(2021)02-0050-05 中图分类号：TP311.52 文献标志码：A

# 基于 Web 的对话应用设计系统

黄 巍，李纪奇，刘国华
（东华大学 计算机科学与技术学院，上海 201620）

**摘 要：** 信息技术领域的客服系统存在人力资源的运用效率低等问题，这给企业带来了大量的维护成本，对企业创新带来了一定的阻碍。因此本文提出一种新型的基于 Web 的对话应用设计系统，不仅提供了基于分支、循环、顺序等设计对话的基础流程控制单元，还包含大量垂直领域内的话题通用组件以辅助在短时间内设计出快速响应、容错率高的场景对话机器人，并提供了深度学习服务用于对话的匹配以及用于提高识别精度的再训练机制，使得对话系统能持续升级和提高对话的效率。本系统使用了先进的开源技术实现，能够快速生成脚本整合进其他现有应用，改善企业客服系统运作效率。

**关键词：** 对话应用设计；对话系统；深度神经网络；Web 应用

## Web-based dialogue application design system

HUANG Wei, LI Jiqi, LIU Guohua
(School of Computer Science and Technology, Donghua University, Shanghai 201620, China)

**【Abstract】** The customer service system in the field of information technology has problems such as low efficiency in the use of human resources, which brings a lot of maintenance costs to enterprises, and brings certain obstacles to enterprise innovation. Therefore, this paper proposes a new Web-based dialogue application design system, which not only provides basic process control units for designing dialogues based on branches, loops, and sequences, but also contains a large number of topical common components in the vertical field to assist in the design in a short time. A scene dialogue robot with fast response and high fault tolerance is provided, and a deep learning service is provided for dialogue matching and a retraining mechanism to improve the recognition accuracy, so that the dialogue system can continue to upgrade and improve the efficiency of dialogue. This system is implemented using advanced open source technology, which can quickly generate scripts and integrate them into other existing applications to improve the operational efficiency of enterprise customer service systems.

**【Key words】** dialogue application design; dialogue system; deep Neural Network; Web application

## 0 引 言

信息技术的发展为企业带来了新的发展机遇，更快的设备和更强大的计算能力驱动企业创造新的产品体验和获取更多的用户，随着产品变得庞杂，对用户使用产品进行引导和解决问题的客服机制也逐渐引起了学界的兴趣。然而客服行业存在的重复率高、响应需求快等特点与人工处理带来的体验产生了冲突。得益于数据量激增以及 GPU 等为图形计算而生的设备的出现为深度学习[1-2]在智能对话领域注入了新的想象力，使得智能对话系统能有效地提高客服领域的运作效率。客服领域分支众多，所以提供能够快速根据需求搭建领域特定的对话系统变得日益重要。

目前，主流的对话系统搭建方式主要有使用提供搭建对话机器人的服务平台如 Google 的 Dialogflow、外包给设计对话机器人的公司或自行研发。分析可知，第一种主要存在定制化能力不强、不易与企业内部系统整合等问题，第二种会存在大量的沟通问题且维护费用也比较可观，第三种的研发成本却十分高昂。

基于此，本文设计并实现了基于 Web 的对话应用设计系统。该系统具备以下特点：提供基于可视化流程图的对话过程设计（如图 1 所示），方便设计和修改对话流程以适应各种客服场景，还提供即时的对话应用测试功能；在初始的深度学习模型上，自动收集与预期情况不符合的答案进行再训练、提供给用户添加额外定义的实体和属性等信息进行训练，持续改进对话效果，让对话更加准确。

**基金项目：** 上海市工业互联网创新发展专项项目“面向纺织服装的行业级工业互联网平台项目”（2019-GYHLW-004）。

**作者简介：** 黄 巍（1996-），男，硕士研究生，主要研究方向：自然语言处理、对话系统；李纪奇（1998-），男，硕士研究生，主要研究方向：Web 开发、工业互联网；刘国华（1966-），男，博士，教授，主要研究方向：外包数据库、隐私保护、文档复制检测等。

**收稿日期：** 2020-11-27

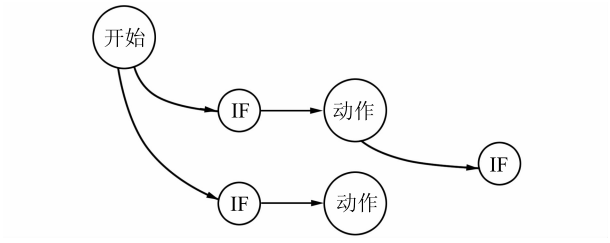


**图 1　可视化流程图的对话过程设计**

**Fig. 1　Design of dialogue process of visual flowchart**

## 1　系统设计

### 1.1　架构设计

系统的整体架构如图 2 所示。本系统使用前后端分离的 B/S 架构[3]进行设计，前端基于 Web 平台，服务器端提供 RESTful API 供前端访问，同时前端还会访问机器学习服务来进行对话过程中的实体、属性的抽取以及对话的匹配等。对此拟展开研究阐述如下。

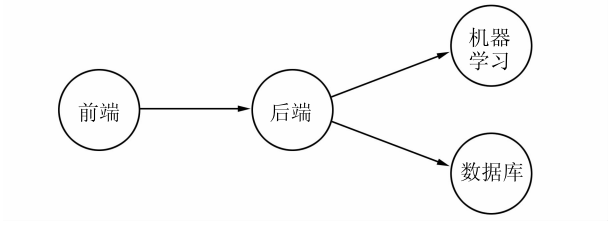


**图 2　系统整体架构图**

**Fig. 2　System overall architecture diagram**

(1)客户端的选择。客户端使用 Web 平台，主要基于 Web 平台具有无平台属性，即在任意系统上如 Mac OS、Linux 或 Windows 上都可以获得一致的访问体验，且 Web 平台技术发展迅速、开发效率高、性能表现良好。

(2)服务器端的选择。服务器端选择了基于 Node. js 的 Koa 框架，这是新一代的 Web 服务器端开发框架，开发效率高，数据库使用 MySQL，同时选用 Sequelize ORM 框架来简化后端与数据库的交互。

(3)机器学习服务的选择。机器学习服务使用注意力机制将对话模型与大规模知识图谱技术相结合，主要包含 3 个模块：编码器-解码器模块、知识解释模块和知识生成模块。其中，知识解释模块采用基于静态的注意力机制[4]，将词向量和检索到的知识图谱图形[5]向量连接起来。知识生成模块采用动态的注意力机制，根据关注权重来读取图中影响选择生成的词语。

(4)数据通信方式选择。数据的通信使用 HTTP 协议，数据的交互主要使用 JSON 格式[6]，这是一种通用的数据交互格式，能被大多数语言所支持。后端和机器学习服务通过 Koa 框架提供 RESTful API 的形式调取使用 MXNet 深度学习框架训练的模型供前端进行访问。

### 1.2　对话匹配算法设计

在设计对话应用过程中，主要通过与用户交互的场景来预设一系列问题和答案，当用户输入的文字与预设的问题匹配，则会选择对应问题的答案返回给用户。为此，本系统设计了一套对话匹配算法。可以选择使用 ML 机器学习的匹配或关键词的匹配，匹配命中的结果主要根据一套优先级规则来确定。算法描述详见如下。

**算法 1　对话匹配算法**

**输入：**置信度 $s$，预设文字 $w_1$，对话文字 $w_2$，深度学习匹配 $m$、关键词匹配 $k$ 和默认匹配 $d$

**输出：**获得胜出的匹配

分别计算 $w_1$ 与 $w_2$ 的深度学习匹配分数 $s(m)$ 和关键词匹配分数 $s(k)$，其中 $s(x) \in [0,1]$；

if $s(m) < 0.7$ and $s(k) = 1$
return　$k$
end if
if $s(m) \geq 0.7$ and $s(k) = 1$
return　$m$
end if
if $s(m) = 0$ and $s(k) = 0$
return $d$
end if

### 1.3　深度学习模型设计

深度学习模型选用了注意力机制，并基于编码器-解码器(seq2seq)模型，将有限的注意力放在重点信息上，从而节省资源，快速获得最有效的信息，而且注意力机制参数少、速度快、效果好。

训练使用的损失函数为交叉熵损失函数，这是更适合衡量 2 个概率分布差异的测量函数。假设训练数据集的样本数为 $n$，交叉熵损失函数[7]定义如式(1)所示：

$$l(\Theta) = \frac{1}{n}\sum_{i=1}^{n} H(y^{(i)}, \hat{y}^{(i)}) , \tag{1}$$

其中，$\Theta$ 表示模型参数。$H(y^{(i)}, \hat{y}^{(i)})$ 函数的定义见式(2)：

$$H(y^{(i)}, \hat{y}^{(i)}) = -\sum_{j=1}^{q} y_j^{(i)} \log \hat{y}_j^{(i)} . \tag{2}$$

即最小化交叉熵损失函数等价于最大化训练数

据集所有标签类别的联合预测概率。

训练使用的是 Adam 优化算法来进行处理,重点是基于训练迭代地更新神经网络权重,适合解决大规模数据和参数的优化问题。

当定义了用户在输入面板中对应的匹配内容后,在对话任务进行过程中,会抽取用户对话中的实体、属性关系等,接着去定义的输入面板内容中进行检索,并匹配相关度高的内容,再对相关度高的内容进行重排序,选择最终的结果来进行后续响应动作的触发。

## 2 系统实现

本系统采用目前最先进的开源技术进行研发,包括前后端及深度学习框架等,同时应用了业界较为热门的数据集和知识库。对此研究中拟做阐析分述如下。

### 2.1 前端实现

前端使用蚂蚁集团开源的插件化的企业级前端应用框架 UmiJS 来实现,其主要的特点是开箱即用,内置路由、构建、部署、测试等完整的开发生命周期。

在前端开发中,先要设计一系列路由界面,将逻辑分发到不同的界面里,以完成需要的功能,UmiJS 内置了 React Router 路由体系,使得开发者只需专注于路由路径及组件逻辑的编写。React Router 拥有简单的 API 与强大的功能,例如代码缓冲加载、动态路由匹配、以及建立正确的位置过渡处理。在编写路由的时候,可通过管理应用的 URL,实现组件的切换和状态的变化,会大量应用在复杂前端应用的开发中,同时也是 React 官方唯一维护的路由框架,社区活跃,新功能迭代快速。

本系统根据具体的场景,设计了包含故事面板页、模板页、故事页、用户输入页、响应用户动作、测试对话系统的运行状态等 6 类页面,前端路由设置详见表 1。

**表 1 前端路由表**

**Tab. 1 Front-end routing table**

| 名称 | 路径 |
|---|---|
| 故事面板页 | /stories/category/all |
| 模板页 | /stories/templates/all |
| 故事页 | /stories/:storyId |
| 用户输入动作页的面板 | /stories/:storyId/ :inId/userSays |
| 响应用户的动作的面板 | /stories/:storyId /:inId/response |
| 测试对话系统面板 | /stories/:storyId/testStory |

前端在开发过程中使用了 Ant Design 组件库来加速开发,开发中主要涉及的就是可视化故事搭建面板的实现和测试故事任务面板的实现。文中对此将给出详述如下。

(1)故事搭建面板的实现。故事搭建面板是一个树状图示,在每个分支的末端可以通过一个加号来扩展对话流程中的响应动作,核心主体采用了 D3.js[8] 来进行流程的可视化展示,其中用到的按钮图标调取自 Iconfont 图标库,内部的核心是维护了一个 JavaScript 数组,通过该数组来记录这个树状流程的关系以及属性等,又通过 D3.js 的树状形状初始化函数来读取这个数组,接着分配对应的属性来进行渲染。研发得到的故事搭建面板则如图 3 所示。


**图 3 故事搭建面板**

**Fig. 3 Story building panel**

(2)动作面板的实现。动作面板主要包含 2 个部分,即:用户输入的匹配和动作的响应。对这两部分的研发探讨具体如下。

① 用户输入的匹配。如图 4 所示。主要是一系列选项,以及针对每个选项有一个下拉框,这个下拉框可以是机器学习式的匹配或者是关键词的匹配,选项列表与下拉框均使用 Ant Design 提供的组件库。机器学习匹配与关键字匹配具体的匹配算法参见前文算法 1。

② 动作响应面板。如图 5 所示。响应的动作是一个顺序流程,先是响应延迟,即在用户输入多久之后回复用户,接着可以添加一系列回复内容,如文本、图片、卡片、轮播图等,还能询问用户问题以获取更多的上下文信息,在点击左方的按钮后,右方会自动添加一个新的虚线以及对应的响应,同样,当用户的输入匹配到了这个动作,在响应中就会额外回复最近添加的一个新的响应动作。

动作响应面板中可以添加额外的人工支持，当机器人无法回答用户的问题时，可以提醒用户进行流转人工操作，获得用户同意后，机器人服务提供方的人工客服会介入处理用户的问题，提升机器人的使用体验。


**图 4　动作面板**

**Fig. 4　Action panel**


**图 5　动作响应面板**

**Fig. 5　Action response panel**

其中，按钮、流程的展示使用 Ant Design 组件库，图标使用了 Iconfont 图标库。

(3)对话面板的实现。通过可视化流程图构建了对话机器人的必要元素后，就可以通过聊天面板测试机器人的运行状况，见图 6。对话面板中，先会触发欢迎动作，发送欢迎语，接着根据用户的选择和回复进行匹配以触发后续的流程，对话面板使用 aurora-imui 来进行方便的搭建，只需传输聊天数据的 JavaScript 数组信息就能够可视化地渲染最终的聊天结果。

## 2.2　后端的实现

后端服务使用 Koa2 实现。Koa2 是 Express 原班人马打造的一个 Web 框架，使用 ES7 的 Async/Await 函数的方式来处理路由操作，相比 Express 而言要更为轻量、健壮，能加速 Web 应用的编码。

数据库使用 MySQL，搭配 Sequelize ORM 框架可以高效地操作数据库。MySQL 是一款优秀的数据库，并且随着技术演变，也在不断地吸纳新的特性以改善使用体验，而 Sequelize 是 JavaScript 的 ORM 框架，开发的友好性使得开发者可以隔离繁琐的原生 SQL 语句操作细节，专注于应用的高效开发。


**图 6　对话面板**

**Fig. 6　Dialogue panel**

后端主要包含用户、故事、动作三张表，对应的 API 路由见表 2。

**表 2　后端路由表**

**Tab. 2　Backend routing table**

| 名称 | 路径 |
|---|---|
| 注册用户 | POST /users/signup |
| 登录用户 | POST /users/login |
| 修改用户资料 | PUT /users/:userId |
| 删除用户 | DELETE /users/:userId |
| 获取用户资料 | GET /users/:userId |
| 添加故事 | POST /stories |
| 获取故事 | GET /stories/:storyId |
| 获取所有故事 | GET /stories |
| 修改故事 | PUT /stories/:storyId |
| 删除故事 | DELETE /stories/:storyId |
| 添加动作 | POST /interactions |
| 获取动作 | GET /interactions/:interactionId |
| 获取所有动作 | GET /interactions |
| 修改动作 | PUT /interactions/:interactionId |
| 删除动作 | DELETE /interactions/:interactionId |

后端根据以上路由列表编写对应的 API，前端使用 Fetch API 发起 HTTP 请求访问后端的路由端口，进行数据的交互。

### 2.3 深度学习服务的实现

深度学习服务使用亚马逊开源的 MXNet 深度学习框架,并使用 Node.js 的 Koa2 框架来提供模型的访问。MXNet 是可扩展的、可以进行快速的模型训练,并灵活支持多种语言,如 C++、Python、Julia、JavaScript 和 Go, MXNet[9] 还可以方便扩展到多个 GPU 来进行训练,也是亚马逊 AWS 的深度学习框架。

模型使用了有 2 层 GRU 结构的编码器-解码器,每一层有 512 个隐藏单元。词嵌入的大小设置为 300,词汇量限制在 10 000 或 30 000,使用 TransE 来获取实体和关系的表示,实体和关系的嵌入大小设置为 100。模型使用以 100 为一个批量的小批量的 Adam 优化器[10],学习率设置为 0.001 5,模型训练的轮次设置为 15 轮次。

通过与领域内 3 个基准模型做比较来验证模型的优势,由此得到:

(1)Seq2Seq[11]模型. 这是一个广泛用于开放领域的对话系统模型。

(2)改编自 Ghazvininejad 等人[2]的知识基础模型 MemNet。存储单元存储 TransE 的知识三元组嵌入。

(3)CioyNet 模型[12]。从知识三元组中复制一个单词或者从词汇表里面生成一个单词。

在模型开发结束后,利用常识知识库[4]和常识对话数据集来进行模型的训练和测试。模型的评价定义了 2 个指标:内容级别的合适性 ppx(回答在语法、主题和逻辑层面是否合适);在给定知识水平下提供的信息性 ent(回答相比已经生成的内容是否提供了更多的信息和知识)。

模型的结果见表 3。表 3 中,分数是模型(CCM)超过其他基准模型的百分比。

**表 3 训练结果分数比对**

**Tab. 3 Training result score comparison**

| 模型 | 整体分数 | |
|---|---|---|
| | *ppx* | *ent* |
| Seq2Seq | 47.01 | 0.713 |
| MemNet | 46.81 | 0.755 |
| CopyNet | 40.21 | 0.930 |
| CCM | 39.12 | 1.174 |

## 3 结束语

本文主要解决针对信息技术客服领域重复劳动率高、人力资源的运用效率低等问题,设计并实现了一个基于 Web 的对话应用设计系统。此系统提供可视化的对话流程设计,以及通过用户输入-对话响应等形式获得高效的对话体验,并且通过深度学习模型来进行用户话语的识别和匹配,提升对话任务的准确性和效率。企业在引进了本文灵活设计对话机器人的系统后,即可提升客服领域解决问题的效率,为企业的发展提供有益的助力。

## 参考文献


[1] LECUN Y, BENGIO Y, HINTON G. Deep learning [J]. Nature, 2015, 521(7553):436.

[2] GHAZVININEJAD M, BROCKETT C, CHANG Mingwei, et al. A knowledge-grounded neural conversation model [C]// Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18). New Orleans, Louisiana, USA:AAAI, 2018:1-9.

[3] HUANG Suping, GUO Xiaojun, LIU Aijun,et al. Application of B/S structure design to the management system for examining construction drawings based on Internet [J]. Journal of Chemical and Pharmaceutical Research, 2014, 6(7):510-520.

[4] ZHOU Hao, YOUNG T, HUANG Minlie, et al. Commonsense knowledge aware conversation generation with graph attention [C]// International Joint Conference on Artificial Intelligence (IJCAI-18). Stockholm, Sweden: Artificial Intelligence Organization,2018:4623-4629.

[5] DUAN Nan. Overview of the NLPCC-ICCPOL 2015 shared task: Open domain Chinese question answering [M]// LI J, JI H, ZHAO D, et al. Natural Language Processing and Chinese Computing. NLPCC 2015. Lecture Notes in Computer Science. Cham: Springer, 2015,9362:562-570.

[6] LV Teng, YAN Ping, HE Weimin. Survey on JSON Data Modelling [J]. Journal of Physics Conference Series, 2018, 1069(1):012101.

[7] HO Y, WOOKEY S. The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling [J]. IEEE Access, 2019, 8:4806-4813.

[8] Bostock M. d3js. D3.js [EB/OL]. [2020]. https://github.com/d3/d3.

[9] GUO Jian, HE He, HE Tong,et al. GluonCV and GluonNLP: Deep learning in computer vision and natural language processing [J]. Journal of Machine Learning Research, 2020, 21:23:1-23:7.

[10] KINGMA D, BA J. Adam: A method for stochastic optimization [J]. arXiv preprint arXiv:1412.6980v8, 2017.

[11] SUTSKEVER I, VINYALS O, LE Q V. Sequence to Sequence learning with neural networks [C]//Proceedings of the 27th International Conference on Neural Information Processing Systems. Quebec,Canada:dblp, 2014,2:3104-3112.

[12] ZHU Xun, LYU Chen, JI Donghong. Keyphrase generation with CopyNet and semantic Web [J]. IEEE Access, 2020, 8:44202-44210.

[13] LI Xiang, TAHERI A, TU Lifu,et al. Commonsense knowledge base completion[C]//Proceedings of the 54th Annunal Meeting of the Association for Computational Linguistics. Berlin,Germany: ACL, 2016:14451455.