# 可行性分析报告



|  |  |
| --- | --- |
| **软件名称：** | **智能对联生成系统** |
| **学院：** | **计算机科学与技术学院** |
| **作者：** | **智能对联生成系统开发小组** |
| **指导教师：** | **张琛** |
| **日期：** | **2020.04.06** |
| **版本：** | **1.1** |

**小组成员**

## 小组成员

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | 分工 |
| 张笑天 | 17030140014 | 组长；后端；机器学习模型 |
| 张震宇 | 17030140013 | 前端 |
| 王重阳 | 17030140020 | 机器学习模型 |
| 王艺静 | 17030140069 | 后端 |

目录

[可行性分析报告 1](#_Toc37609807)

[小组成员 2](#_Toc37609808)

[1. 引言 4](#_Toc37609809)

[a) 标识 4](#_Toc37609810)

[b) 背景 4](#_Toc37609811)

[c) 项目概述 4](#_Toc37609812)

[d) 文档概述 4](#_Toc37609813)

[2. 引用文件 4](#_Toc37609814)

[3. 可行性分析的前提 4](#_Toc37609815)

[a) 项目的要求 4](#_Toc37609816)

[b) 项目的目标 5](#_Toc37609817)

[c) 项目的环境、条件、假定和限制 5](#_Toc37609818)

[d) 可行性分析方法 6](#_Toc37609819)

[4. 可选的方案 6](#_Toc37609820)

[a) 无原有方案 6](#_Toc37609821)

[b) 可重用系统 6](#_Toc37609822)

[c) 可选择系统方案1 6](#_Toc37609823)

[d) 可选择系统方案2 6](#_Toc37609824)

[e) 最终方案准则 6](#_Toc37609825)

[5. 所建议的系统 6](#_Toc37609826)

[a) 说明 6](#_Toc37609827)

[b) 数据流程和处理流程 6](#_Toc37609828)

[c) 与原系统的比较 9](#_Toc37609829)

[d) 要求 9](#_Toc37609830)

[e) 局限性 9](#_Toc37609831)

[6. 经济可行性 9](#_Toc37609832)

[a) 投资 9](#_Toc37609833)

[b) 预期的经济收益 9](#_Toc37609834)

[c) 市场预测 9](#_Toc37609835)

[7. 技术可行性 10](#_Toc37609836)

[8. 法律可行性 10](#_Toc37609837)

[9. 用户使用可行性 10](#_Toc37609838)

[10. 未来与项目有关的问题 10](#_Toc37609839)

[11. 注解 10](#_Toc37609840)

**正文**

## 引言

### 标识

软件标题：智能对联生成系统

版本号：v1.0

### 背景

提出条件：本软件为西安电子科技大学计算机科学与技术学院软件工程课程的作业。该作业要求规范地实现一个软件。小组根据该要求订立了开发智能对联系统的目标，因为该系统同时兼顾目前流行的自然语言处理领域和互联网开发领域。

要求：实现一个智能对联系统，可以根据上联智能生成下联。

目标：使得用户可以输入上联得到下联，起到丰富精神生活作用。

实现环境：阿里云服务器 Ubuntu 18.04

限制条件：开发者机器性能有限

### 项目概述

使用项目：智能对联生成系统

软件用途：通过网页端可以获取到根据已有上联只能生成的下联。

开发历史：本项目未曾有前置版本。但在服务器搭建，Tensorflow使用上已有经验。

投资方：开发小组自费

需方：西安电子科技大学计算机科学与技术学院软件工程课程

用户：网页使用者

开发方：开发小组，成员：张笑天，王重阳，王艺静，张震宇

支持机构：西安电子科技大学计算机科学与技术学院软件工程

当前运行现场：虚拟机VMWare中Ubuntu 19.10，Windows 10平台Anaconda

计划运行现场：阿里云 Ubuntu 18.04

### 文档概述

本文档为项目 智能对联生成系统 的可行性分析报告，用于论证与确定该项目是否可行。

保密性：该文档可以公开于网络，但应注意项目以及该文档本身均遵循GPLv3协议开源。

<http://www.gnu.org/licenses/quick-guide-gplv3.html>

## 引用文件

GPLv3协议：

<http://www.gnu.org/licenses/quick-guide-gplv3.html>

## 可行性分析的前提

### 项目的要求

功能：用户可以输入自定义的上联得到下联。

性能：用户得到输出速度要足够快。

系统输入：用户自身的上联

系统输出：系统通过NLP模型得到的下联

健壮性：系统应该能够甄别用户输入是否合法，并能够恰当处理非法输入。系统在发生异常时应该尽量保证系统不关闭。

可移植性：系统应不修改或者较少修改即可一直到其他平台。

可拓展性：系统应该易于扩展其他功能。

美观性：网页设计应该尽量美观

完成期限：2020.06.01前

### 项目的目标

使得用户可以在网页上方便地输入自自定义的上联得到下联，起到丰富精神生活作用。

### 项目的环境、条件、假定和限制

硬件：

开发硬件使用X86架构系统，CPU包括Intel Core i5 8250u，AMD Ryzen 2600X，GPU包括Nvidia GTX 1050Ti，Nvidia GeForce MX150，Nvidia GTX 1060。

实际运行硬件使用阿里云的云服务器ECS。

实例：共享计算型 n4系列 III1核 2GB

I/O 优化实例：I/O 优化实例

系统盘：高效云盘/dev/xvda40GB

带宽：1Mbps按固定带宽

CPU：1核

可用区：华东 2 可用区 B

操作系统：Linux64位Ubuntu 18.04 64位

内存：2GB

地域：华东 2

网络类型：专有网络

软件：

开发软件包括

系统：Windows 10， Ubuntu 19.10，Ubuntu 18.04

虚拟机：VMWare Workstation Pro，

解释器：Python 3.7

环境：Anaconda

机器学习框架：Tensorflow 1.14

IDE：PyCharm Community，IDLE，Visual Studio

编辑器：Visual Studio Code，Vim

浏览器：Google Chrome，Firefox，Microsoft Edge

版本控制：Git，GitHub

生产环境软件包括

系统：Ubuntu 18.04

解释器：Python 3.7

环境：Anaconda

机器学习框架：Tensorflow 1.14

代理服务器：Nginx

Web服务器：uWsgi

Web应用框架：Django

开发限制：开发周期短，小组成员能力限制，开发使用硬件性能限制。

### 可行性分析方法

复查系统规模和目标

研究目前正在使用的系统

导出新系统的高层逻辑模型

重新定义问题

导出和评价供选择的解法

推荐行动方针

## 可选的方案

### 无原有方案

### 可重用系统

之前移动互联网导论搭建的Django后端

差距：该Django后端用于点餐，需要另加App

### 可选择系统方案1

前端+Django后端+Tensorflow训练

### 可选择系统方案2

前端+Flask后端+PyTorch训练

### 最终方案准则

可重用；资源较多。

尽管Flask较简单，但是可复用Django因此使用Django；尽管PyTorch相对更简单，但是Tensorflow资料相对较多，因此最终选用。

## 所建议的系统

### 说明

前端：HTML5+CSS+JavaScript

因为网页为静态网页，因此使用简单的前端系统

后端：Nginx+uWsgi+Django

使用之前复用的Django作为后端服务器框架

深度学习框架：

Python3+Ananconda+Tensorflow+Tensor2Tensor，使用序列到序列模型

使用较为流行的Tensorflow作为学习框架，使用T2T深度学习模型库

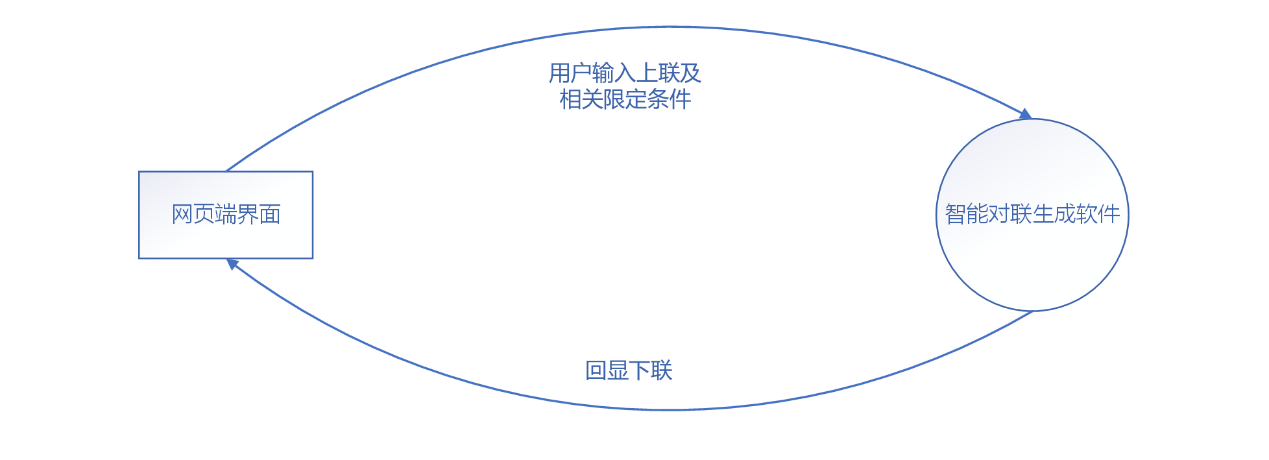
用户通过在网页端输入上联，通过HTTP协议发送到后端。同样，后端得到下联以后，通过HTTP返回到前端给用户。

后端得到上联以后，通过与Tensorflow模型的API进行通信，将后端发送给模型，并且得到其回复。

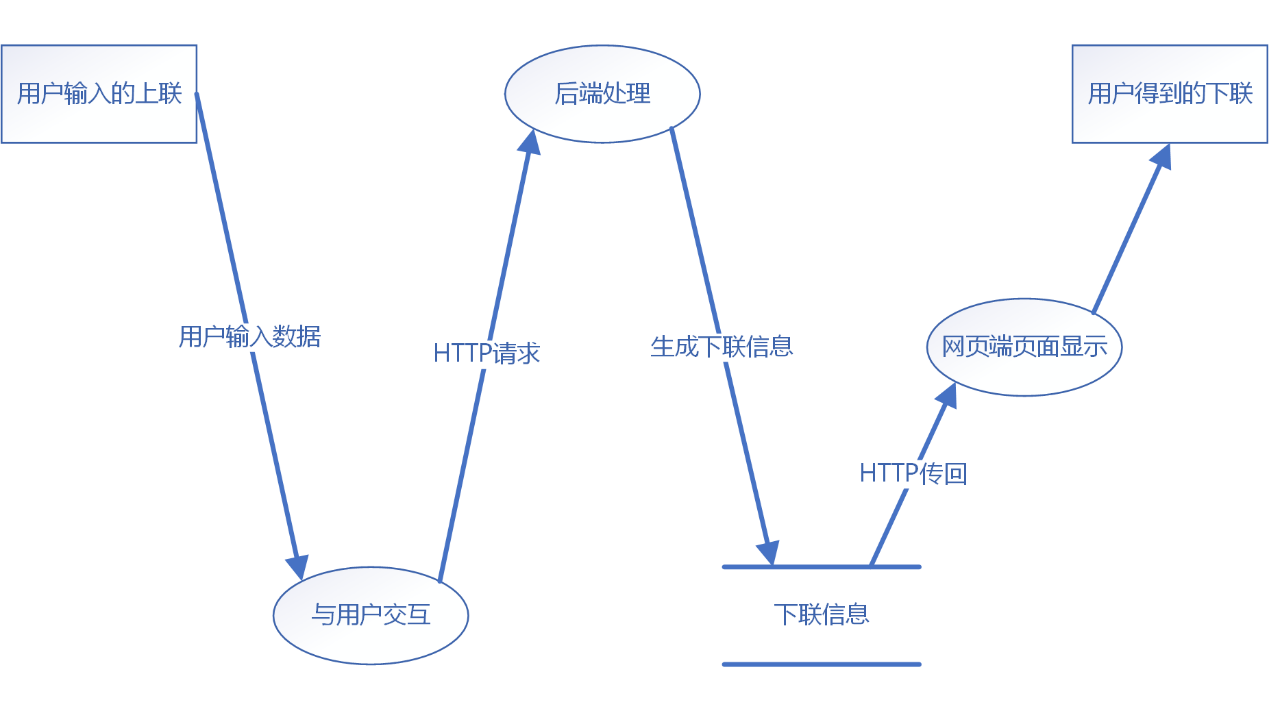
在模型中，上联通过编码器Encoder编码为向量的形式，再通过解码器Decoder得到下联，最后通过API返回到后端服务器。

### 数据流程和处理流程

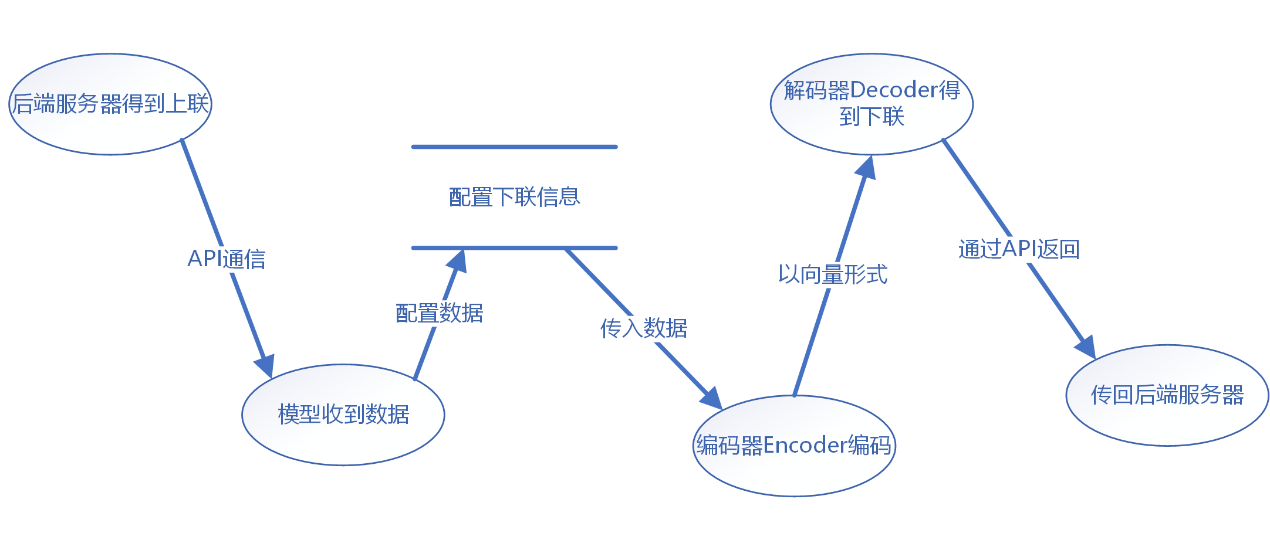
总体：



第二层，网络通信部分：



第二层，后端部分：



数据字典：

用户输入的上联

|  |  |
| --- | --- |
| 名字 | 用户输入的上联 |
| 分类 | 数据元素 |
| 描述 | 在前端由用户输入的上联 |
| 定义 | 用户输入的上联={utf-8汉字}7 |
| 位置 | 用户->前端 |

用户得到的下联

|  |  |
| --- | --- |
| 名字 | 用户得到的下联 |
| 分类 | 数据元素 |
| 描述 | 在前端由显示给用户的下联 |
| 定义 | 用户得到的下联={utf-8汉字}7 |
| 位置 | 前端->用户 |

前端传送到后端的上联

|  |  |
| --- | --- |
| 名字 | 前端传送到后端的上联 |
| 分类 | 数据流 |
| 描述 | 前端通过HTTP传送到后端的上联 |
| 定义 | 前端传送到后端的上联={HTTP规定} |
| 位置 | 前端->后端 |

后端传送到前端的下联

|  |  |
| --- | --- |
| 名字 | 后端传送到前端的下联 |
| 分类 | 数据流 |
| 描述 | 后端通过HTTP返回到前端的下联 |
| 定义 | 后端传送到前端的下联={HTTP规定} |
| 位置 | 后端->前端 |

后端传送到机器学习模型的上联

|  |  |
| --- | --- |
| 名字 | 后端传送到机器学习模型的上联 |
| 分类 | 数据流 |
| 描述 | 后端通过API将上联给机器学习模型 |
| 定义 | 后端传送到机器学习模型的上联={API规定} |
| 位置 | 后端服务器->机器学习模型 |

机器学习模型传送到后端的下联

|  |  |
| --- | --- |
| 名字 | 机器学习模型传送到后端的下联 |
| 分类 | 数据流 |
| 描述 | 机器学习模型通过API传送给后端的下联 |
| 定义 | 机器学习模型传送到后端的下联={API规定} |
| 位置 | 机器学习模型->后端服务器 |

语义编码向量

|  |  |
| --- | --- |
| 名字 | 语义编码向量 |
| 分类 | 加工 |
| 描述 | 从Encoder生成到Decoder被解码 |
| 定义 | 语义编码向量={Encoder规定} |
| 位置 | Encoder->Decoder |

### 与原系统的比较

无原系统

### 要求

设备：一台云服务器ECS

软件：Linux操作系统，Nginx，Django，Python3解释器，Anaconda

运行：满足设备与软件要求即可

开发：Python与前端的开发工具，Python包管理工具即Anaconda，训练模型需要GPU

环境：无

经费：每月20人民币用于维护服务器与域名，2000人民币一次性投资用于升级GPU

### 局限性

没有好的用户纠错机制。

受制于服务器性能，生成可能较慢。

没有上联的生成系统。

## 经济可行性

### 投资

基本建设投资：

GPU购买：一次性2000人民币，Nvidia Turing GPU

硬件投资：每月10元购买云服务器，10元购买域名

### 预期的经济收益

无收益

### 市场预测

对于这类简单娱乐软件需求未来将呈平稳趋势

## 技术可行性

本团队目前人员有后端与机器学习开发经验。但前端开发经验不足。因为前端功能相对简单，所以如果出现前端无法完成情况，可以通过降低前端界面要求补救。

目前本团队已经购置服务器与域名，并有性能一般的GPU使用，因此硬件条件充足。

技术条件，目前对于自然语言处理的研究和前后端的实践已经较为充分，因此技术条件良好。

NLP：基于Attention机制，该机制训练效果好，时间短，在自然语言翻译中大量使用。

后端：Django，Python最常用的开源框架之一。

前端：技术成熟，要求简单。

## 法律可行性

本系统软件与开发主要使用开源框架，因此法律上一致遵循GPLv3开源协议，避免对于触犯开源协议的风险。

本系统使用阿里云服务器与域名，因此根据中华人民共和国法律法规进行备案，同样运营也遵循中华人民共和国法律法规。

## 用户使用可行性

本系统的用户使用的功能相对简单，用户只需要了解基本的网页操作即可。

## 未来与项目有关的问题

未来取决于项目进度添加需求，例如客户端应用，通过图片联想上联等。

## 注解

NLP：自然语言处理

云服务器（ECS）: 一种简单高效、安全可靠、处理能力可弹性伸缩的计算服务

CPU：中央处理器，计算机系统的运算和控制核心，是信息处理、程序运行的最终执行单元

GPU：图形处理器，一种专门在个人电脑、工作站、游戏机和一些移动设备上做图像和图形相关运算工作的微处理器。

X86：一系列基于Intel 8086且向后兼容的中央处理器指令集架构

Linux：一套免费使用和自由传播的类UNIX操作系统。

Ubuntu：一个以桌面应用为主的Linux操作系统。

Windows：美国微软公司研发的一套操作系统

Python：一种跨平台的计算机程序设计语言

VMware Workstation：一款功能强大的桌面虚拟计算机软件，提供用户可在单一的桌面上同时运行不同的操作系统

Anaconda：开源的Python包管理器

Tensorflow：TensorFlow™是一个基于数据流编程的符号数学系统

PyTorch：是一个开源的Python机器学习库

GPL：GNU General Public License，GNU通用公共许可证

IDE：集成开发环境，Integrated Development Environment是用于提供程序开发环境的应用程序，一般包括代码编辑器、编译器、调试器和图形用户界面等工具

PyCharm：一种Python集成开发工具

IDLE：Integrated Development and Learning Environment，集成开发和学习环境，是Python的集成开发环境

Visual Studio：美国微软公司的开发工具包系列产品

Visual Studio Code：跨平台源代码编辑器

Vim：功能强大、高度可定制的文本编辑器

Google Chrome：是一款由Google公司开发的网页浏览器

Firefox：是一个自由及开放源代码的网页浏览器

Microsoft Edge：是一款由Microsoft公司开发的网页浏览器

Git：一个开源的分布式版本控制系统

GitHub：一个面向开源及私有软件项目的托管平台

Nginx：一个高性能的HTTP和反向代理web服务器

uWsgi：一个Web服务器与web应用通信的规范实现

Django：一个开放源代码的Web应用框架

Flask：一个使用 Python 编写的轻量级 Web 应用框架

HTML5：超文本标记语言5，构建Web内容的一种语言描述方式

CSS：层叠样式表，Cascading Style Sheets是一种用来表现HTML或XML等文件样式的计算机语言

JavaScript：是一种具有函数优先的轻量级，解释型或即时编译型的编程语言

Tensor2Tensor：一套基于TensorFlow的深度学习系统