1. INTRODUCTION

1.1. Directory introduction

There is a README.md document under the root directory to help you understand the project easily, and there are three subfolders, Miscellaneous, Documentation, and Code.

In the Documentation directory, the final report, user guide, and installation guide are included. In the Miscellaneous directory, we include download links to the final model files, training dataset, and other necessary data.

In the Code directory, there are two subdirectories. The Engineering directory is the code required for the engineering system to run. The Algorithm directory is the training code for all modules.

1.2. Engineering introduction

The system runs in the browser, and the users can access our chatbot by visiting the web page. Although the system is designed to run in various browsers, we recommend using the chrome browser on the PC side for access.

The system adopts B/S (browser/server) architecture, which has two subsystems included, the front-end and the back-end. The front-end system implements the user interface and interactive functions of the chatbot. The back-end system implements all the API services and model inference.

In the submitted code package, the compiled product of the front-end system has been included in the static directory of the backend code, **so if you do not need to modify the front-end code, you only need to run the backend system to use the chatbot**.

Please follow the next section of the backend installation guide to run the system.

2. ENVIRONMENT INSTALLATION

2.1. Python installation

Please follow the instructions and install the python environment by yourself first. The best practice is to create an isolated env for the project, but it's not mandatory.

The recommended python version is **3.7.11**.

2.2. Dependencies installation

Go to the **Code** directory and run the following command to install the project dependencies. This command will help you install all the libraries needed for the project.

pip install -r requirements.txt

If "pip command not found" appears, please check your python and pip environment. If "requirements.txt does not exist" appears, please check if your current directory is correct.

3. ALGORITHM FOLDER

In the algorithm directory, there are three subdirectories containing three function-related algorithm files. The data processing, model training, and model evaluation code for each of our models are listed under each subdirectory.

4. ENGINEERING FOLDER

4.1. Back-end

4.1.1. Model files download

Because the model files are too large, they are not packaged in the compressed package. Please visit the google drive share link to download all files one by one and put them into the backend/models/ directory. Note: 'fake_model_attention' must be a file not zipped folder.

4.1.2. Google cloud sdk download

Because google translation's basic service is conditionally free, please use your own key and set all configure by your own key. Please follow this document.

4.1.3. Run the service

Run the following command in the console to export an environment variable.

For Bash Users:

> **export FLASK_APP=app**

> **flask run**

For CMD Users:

> **set FLASK_ENV=development**

> **flask run**

For Powershell Users:

> **$env:FLASK_ENV = "development"**

> **flask run**

Now you can see the project is running smoothly in the console. Access the service link prompted in the console in your browser (by default is http://127.0.0.1:5000/) and you can use the chatbot!

4.2. Front-end

Although it is not necessary to run a separate front-end system, you can already use chatbots smoothly. But if you want to make changes to the front-end code, or want to see how it was developed, you can follow the instructions below to run the front-end system alone.

4.2.1. Environment installation

Please follow the instructions and install the Node.js environment by yourself first.

4.2.2. Dependencies installation

Go to the frontend code directory and run the following command to install the project dependencies. This command will help you install all the libraries needed for the project.

npm install

If "npm command not found" appears, please check your node and npm environment.

4.2.3. Run the project

Run the following command in the console to export an environment variable.

npm run serve

Now you can see the front-end system is running smoothly in the console. Access the service link prompted in the console in your browser (by default is http://localhost:8080/) and you can see the chatbot now!

It should be noted that only the front-end system is running on port 8080, and all back-end requests will be forwarded to the service on port 5000 through the proxy settings in the vue.config.js file. If your backend system is not running on port 5000, you need to modify this configuration.