Final Project Report

Project Title:

Text Translation App Using Youdao API

高泽华 Robin

Class number: 35

Course name:

Optimization Method

Department of Software Engineering

Hebei University

Declaration

This report has been prepared on the basis of our own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 9902 words

Student Name: 高泽华

高泽华 Robin 20202504073

Date of Submission: Dec 7, 2022

Table of Contents

Abstract

As one of the popular categories of artificial intelligence hardware, although many companies are researching and developing intelligent translators, due to the relatively fixed specific use scenarios, in a small number of fields, such as some text translation application scenarios in ordinary offices, it is still An online translation site is required.

In order to meet the needs of personal translation, and also to test whether I have the ability to design and develop complete projects, I use python programming language combined with QT library to develop windows and integrate Youdao dictionary API, and finally package it into an efficient translation software. To achieve translation needs. It basically meets the standard of use and has good scalability.

The translation interface of this project adopts Youdao, a well-known international company. The feature of this interface is that it supports mutual translation of 104 languages, and focuses on the translation of science and technology and academics.

The overall development of the software follows the modular development method, and through repeated testing, the effect is achieved to achieve the desired goal.

Keyboard:text，translator，API，python，QT，python library

Chapter 1: **Introduction**

The main function of this translation app is to provide word definition and online translation. This translation app can help you look up the meaning of words and solve difficult problems encountered in life. In addition, it can also help you translate foreign languages, provide text automatic detection language, and support multiple languages, such as English, Japanese, Korean, French, etc. Another handy feature of this translation software is that it can be used offline, which means you can use it without an internet connection. In short, this translation app is a very practical tool that can help you improve your language skills and improve your reading comprehension.

The function of text translation software can realize lightweight translation software design. This program uses Youdao Dictionary API to translate the text to be translated and can automatically detect the language to be translated. It accepts the text to be translated as input and uses Youdao Dictionary API to translate the text. Finally, it prints the translation result. At the same time this software has a user interface that allows the user to select the language to be translated and the text provided. This software supports translation in multiple languages or provides translation suggestions,

The development design is divided into three major modules,

First call the library first, the interface is designed with the QT library, and a clear translation software window is designed

Then call the Youdao dictionary API to write the interface function about any language --> Chinese,

Write an interface function about any language --> English,

The interface for translating into three languages is written above. If you want to translate from one language to another, you only need to modify the abbreviation below, which can automatically identify the language that needs to be translated and quickly translate

The last is to call the main functions of these interfaces to realize the text translation function.

Chapter 2: **Background research**

Translation software development is the process of creating software that is able to translate text from one language to another. This is typically done by using machine learning algorithms that are trained on large datasets of human-translated texts. The goal of translation software is to produce translations that are accurate, fluent, and natural-sounding, so that they can be used in a variety of settings, from personal communication to business and government applications.

Translation software uses a combination of rule-based and statistical approaches to perform the translation. In rule-based translation, the software uses a set of pre-defined rules that tell it how to translate words and phrases from one language to another. These rules are typically based on the grammar and syntax of the languages involved, as well as on any specialized vocabulary that is relevant to the text being translated.

Statistical translation, on the other hand, uses machine learning algorithms to learn how to translate text from one language to another. These algorithms are trained on large datasets of human-translated texts, which are used to teach the software how to produce accurate and fluent translations. The algorithms are able to learn patterns in the text and use them to generate translations that are more accurate and natural-sounding than those produced by rule-based systems.

The development of translation software requires a deep understanding of both linguistics and computer science. Linguists work on developing the rules and algorithms that are used by the software, while computer scientists work on implementing these rules and algorithms in a way that is efficient and effective. The development process also involves extensive testing and evaluation to ensure that the software is able to produce high-quality translations.

In recent years, with the popularity of the concept of artificial intelligence, many first-class companies in the world have invested huge sums of money in it, and many start-up companies have also taken artificial intelligence as their main business direction. Three crazes swept the world.

In fact, in 2014, artificial intelligence expert Ng Enda joined Baidu, and Hawking and Musk put forward the "artificial intelligence demon theory". The scene is not clear, so it fell into silence in 2015.

However, under the temporarily calm surface of artificial intelligence, there are hidden waves. With the rapid development of Internet technology, the huge improvement of computer performance, the further development of big data and cloud computing, all of these have accumulated potential energy for the third climax of artificial intelligence. AlphaGo was born in 2016, and the wave of artificial intelligence swept the world.

In 2017, Wuzhen Think Tank, Phoenix Technology, Xinhuanet, Beijing News, Southern Metropolis Daily, Xtecher and other authoritative media jointly released a series of reports "Wuzhen Index: Global Artificial Intelligence Development Report (2017)". The conclusion of the report is: Although China continues to catch up with the United States in the number of artificial intelligence companies, but the United States still dominates the world in terms of the scale of artificial intelligence financing, accounting for more than 60%.

The problems that have always existed in the field of online translation are low accuracy, low efficiency, and grammatical errors. With the development of artificial intelligence, artificial intelligence translation is becoming more and more popular, and now the accuracy of Youdao translation is getting higher and higher.

Therefore, I have adopted the Youdao translation interface, which can automatically identify the language that needs to be translated and translate quickly, achieving the same level of translation as Google.

Text translation is a technology that allows users to translate a piece of text from one language to another. This technology can be used in many different application scenarios, including supporting the business operations of multinational companies, helping people understand text information in different languages, and improving the user experience of multilingual websites, etc.

Before designing a text translator, there are several factors to consider:

Supported languages: First, you need to determine which languages are supported by the text translator, and the translation between them.

Translation Quality: The quality of text translations is a very important factor, as users expect translations that are as accurate and fluent as possible. When designing a text translator, you need to consider how to improve translation quality.

User Interface: A text translator needs to have a user-friendly interface that makes it easy for users to use it for translation.

Performance: The text translator needs to have good performance, be able to translate quickly, and be able to withstand a large number of concurrent requests.

Scalability: A text translator should be scalable to accommodate growing numbers of users and languages.

2.2 design method

The basic idea behind designing a translation software is to translate input text from one language to another. To accomplish this, translation software typically uses natural language processing techniques to understand the incoming text and machine translation techniques to translate the text into the target language. Machine translation techniques may use neural network models to generate high-quality translations, or they may use rule-based translation methods to translate text. Translation software may also use language models to ensure that translated text is both syntactically and semantically correct.

Overall, the main goal of designing a translation software is to achieve high-quality translation from one language to another using natural language processing and machine translation techniques.

Python is a popular programming language that can be used to design translation software. Python has a wealth of third-party libraries and tools that can help developers implement natural language processing and machine translation capabilities.

To design a translation software, we can use Python's natural language processing library to process the input text. For example, we can use the NLTK library for word segmentation, part-of-speech tagging, and syntactic analysis to understand the grammatical structure and semantic meaning of the input text.

Next, we can use Python's machine translation library to implement the translation function. For example, we can use the Google Translate API to achieve automatic translation, or use the OpenNMT library to build a neural network model to generate high-quality translations.

Finally, we can use Python's language model library to ensure that the translated text is both syntactically and semantically correct. For example, we can use the Keras library to train a language model to predict whether the translated text conforms to the language rules.

In general, Python can be used as an excellent programming language to design translation software, because it provides a wealth of natural language processing, machine translation and language model tools, which can help developers achieve high-quality translation

First of all, Python has many excellent third-party libraries and tools that can be used to design translation software. For example, NLTK is a popular natural language processing library that provides word segmentation, part-of-speech tagging, syntactic analysis, and other features that can help developers understand the syntax and semantics of input text. Google Translate API and OpenNMT are machine translation libraries that can be used to implement automatic translation and neural network model translation. Keras is a deep learning library that can be used to train language models.

Secondly, Python's concise and easy-to-use syntax is also its advantage as a translation software design language. The syntax of Python is simple and clear, and it is easy to learn. For developers, it can save a lot of time and effort, and it is easier to realize complex translation functions.

Finally, Python also has extensive community support and resources to help developers solve problems they encounter. Python has many enthusiastic users and developers, who provide rich documentation, code samples, and technical support in the community, which can help developers solve problems quickly and achieve high-quality translation software

Overall, Python is a very suitable programming language for designing translation software. It has rich natural language processing, machine translation and language model tools that can help developers achieve high-quality translation functions. In addition, Python's concise and easy-to-use syntax and extensive community support are also its advantages as a translation software design language. Therefore, using Python to design translation software is a good choice

With the development of technology, people all over the world can communicate with each other despite language barriers. One of the tools that can help bridge this gap is machine translation. With the help of machine translation, it is now possible to translate text from one language to another using specialized software.

In this paper, we will focus on text translation using Youdao Dictionary API. Youdao Dictionary API is a set of application programming interfaces (APIs) developed by NetEase that allow communication with and integration of NetEase services into other services. One API of Youdao Dictionary API collection is Youdao Dictionary Translation API, which allows developers to integrate the text translation function of Youdao Dictionary Translation into their own applications.

In this study, we will discuss the various features and functions of Youdao Dictionary Translation API, and how it can be used to develop text translation applications. We will also discuss various challenges and limitations of using machine translation, and how to address them.

The importance of text translation:

The ability to communicate with people all over the world is becoming increasingly important in the globalized world we live in today. People who do not speak the same language can now communicate with each other with the help of text translation.

Translation has a wide range of applications in various fields. For example, in the field of international business, text translation allows companies to communicate with foreign clients and partners. It also enables them to translate documents and contracts into different languages, making it easier to do business with people from all over the world.

In education, text translation allows students and teachers to access information and resources in different languages. This is especially important in a world where access to information is critical to learning and personal development.

Furthermore, text translation also has important applications in healthcare. With the help of text translation, doctors and healthcare professionals can communicate with patients who do not speak the same language, ensuring they receive the necessary care and treatment.

Overall, the ability to translate text from one language to another is critical to facilitating communication and understanding between people from all over the world.

Youdao Dictionary Translation API:

Youdao Dictionary Translation API is part of Youdao Dictionary API Collection, which allows developers to integrate the text translation function of Youdao Dictionary Translation into their own applications.

The API uses neural machine translation, a type of artificial intelligence that allows text to be translated from one language to another.

Youdao Dictionary Translation API provides many features and functions that make it an effective text translation tool. Some of these features include:

1. Support multiple languages: Youdao Dictionary translation API supports translation between more than 100 languages. This makes it a suitable text translation tool to use in many situations where communication in different languages is required.

2. High translation quality: Youdao Dictionary translation API uses neural machine translation, which is known for its high quality. This means that the translations generated by the API are accurate and close to human translations.

3. Customization options: Youdao Dictionary Translation API allows developers to customize the translations generated by the API according to their own needs. For example, developers can specify the target language for translation and the format of the translated text.

4. Easy to integrate: Youdao Dictionary translation API is easy to integrate into existing applications. This means that developers can easily add text translation functionality to existing applications without having to develop translation algorithms.

Use Youdao Dictionary Translation API to develop text translation applications:

To use the Youdao Dictionary Translation API to develop text translation applications, you first need to register the Youdao Toolkit (SDK) to develop applications.

When developing a text translation application, developers need to consider how to obtain the text to be translated and how to display the translated text in the application. For example, an application might provide a text box that allows users to enter text to be translated, and then use Youdao Dictionary Translation API to translate the text into the desired language.

In addition, developers also need to consider how to handle network connectivity issues. Since the text translation application needs to connect to Youdao Dictionary Translation API through the Internet, it must be considered that the network connection is interrupted or unavailable. In this case, the application may need to provide some way to notify the user of network connectivity issues and retry the translation when the network is restored.

Chapter 3: **Design & Implementation**

3.1 System design

In order to design a text translation software with forms, we need to consider the following aspects:

Form layout: We need to design the layout of the form in order to place the text input box, translation button and translation results.

Component Layout: We need to design the layout of the text input box, translation button and translation results so that the form looks nice and easy to use.

Interaction: We need to design the interaction between the text input box, the translation button and the translation result, so that when the user enters text and clicks the translation button, the translation can be performed correctly.

Youdao Dictionary API: We need to use Youdao Dictionary API to realize the function of text translation.

Taking these aspects into consideration, we can design the following form:

form design

In this form, we use a vertical layout to place the text input box, translation button and translation results. The text input box and translation button are at the top of the form, and the translation results are at the bottom of the form.

Both the text input box and the translation button have sufficient width and height to allow users to easily enter text and click the button to translate

In summary， we have：

Needs analysis: develop a tool that can translate multiple national languages to improve learning efficiency

Code writing: choose to use python language 3.8, tool pycharm. Form implementation generated from reading ui file 'window_2.ui' Get the translation result by calling Youdao API interface. Due to some unstable factors in some domestic servers in China and the Google translation api is disabled in China, I chose to adjust and use Youdao Translation, a well-known translation software api in China, to make an alternative attempt.

Interface design: Created by: PyQt5 UI code generator 5.14.1

Use python designer/qt to design the program interface, mainly with translation function display, to achieve a simple and smooth interface

Combine the code and the designed interface together to package and generate an exe executable file, and the development is complete

Software testing, repeatedly debug the developed program, if any problem is found, repeat steps 2 to 5

3.2 System implementation

3.2.1 System implementation

First design some libraries needed for development, such as sys, PyQt5, json, mrandom, urllib, http.client, uuid, requests, hashlib, time, imp, etc.

PyQt5 is a software library for creating graphical user interfaces (GUIs). It is based on the Qt library, a development framework for creating cross-platform applications. PyQt5 provides a Qt-like API that allows users to write GUI applications in the Python language. It also provides many useful functions such as signals and slots, event handling, layout management and many tools that can be used to create modern applications. Use PyQt5 to create powerful GUI applications quickly and easily.

http.client is a module in the Python standard library that provides a set of classes and methods for making HTTP requests and handling responses. It supports various types of HTTP requests, such as GET, POST, PUT, DELETE, and HEAD, and provides various useful features, such as automatically handling encoding, authentication, cookies, and redirection. Use the http.client module to quickly and easily make HTTP requests and process response data. It is the tool of choice for developing web applications and crawlers.

sys is a standard library of the Python language, which provides many functions and variables for interacting with the Python interpreter. For example, it can be used to access command-line arguments, obtain program version information, access environment variables, and implement custom exception handling. It also provides functions for interacting with the operating system, such as reading and writing files and directories, executing subprocesses, and terminating programs. By using the sys library, you can easily extend the functionality of the Python interpreter and interact with the operating system.

JSON (JavaScript Object Notation) is a format for storing and exchanging text data. It is a subset of the JavaScript language and is very readable. It can be used to represent various types of data, such as numbers, strings, Boolean values, arrays, and objects, as well as complex nested relationships between them. In Python, the json library provides a set of functions and methods for working with JSON data. It can be used to read, write, and parse JSON data, as well as convert between Python objects and JSON data. By using the json library, you can easily use JSON data in Python programs.

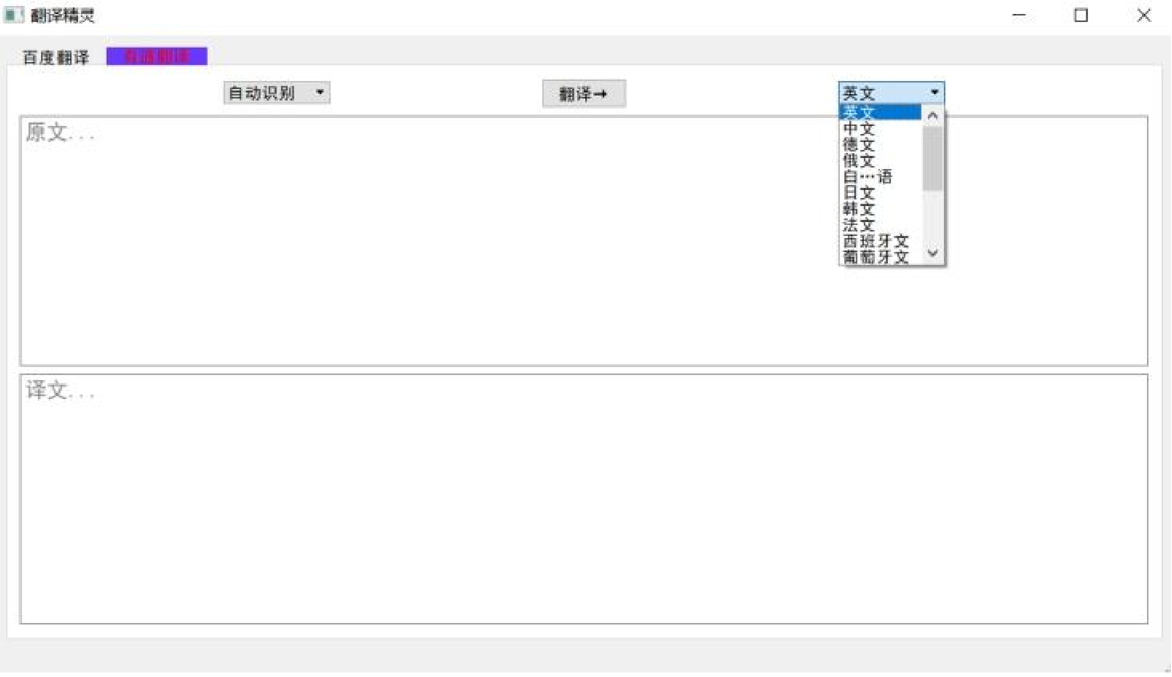
Requests is a Python third-party library that provides a set of simple and easy-to-use APIs for making HTTP requests. It supports various types of HTTP requests, such as GET, POST, PUT, DELETE, and HEAD, and provides many useful features, such as automatically handling encoding, authentication, cookies, redirection, and file uploads. Use Requests to quickly and easily send HTTP requests and get response data. It is the tool of choice for developing web applications and crawlers.

hashlib is a module in the Python standard library that provides a set of classes and methods for computing message digests. Message digest is an encryption technology that generates a fixed-length digital signature by hashing the input data to verify the integrity and security of the data. The hashlib module supports multiple hash algorithms, such as MD5, SHA1, SHA256, and SHA512, and provides some useful tools, such as hash calculation, HMAC calculation, and password hashing. The hashlib module can be used to conveniently implement the message digest function in the Python program.

code show as below

```
import sys
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import *
import json
import random
from urllib import parse
import http.client
import uuid
import requests
import hashlib
import time
from imp import reload
```

- Translation software interface



- Fig 3-1 Translation software interface

The interface design is as follows, the window is divided into two parts, the upper part is the input layer, and the lower part is the receiving window for the output result.

Among them, the above form is from left to right to automatically recognize the language, translate the button and select the relevant language of the output

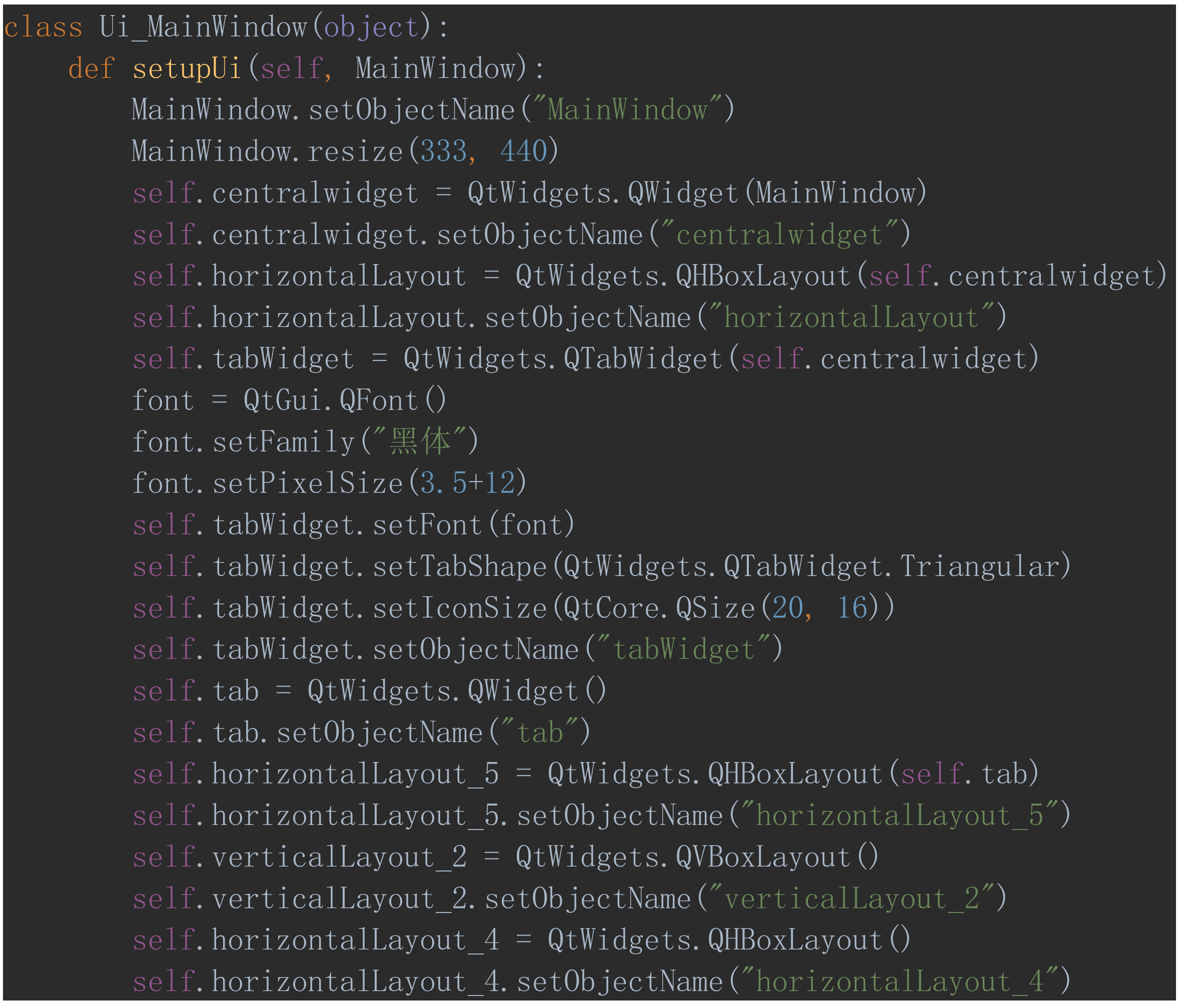
Create and design an interface form

Created by: PyQt5 UI code generator 5.14.1

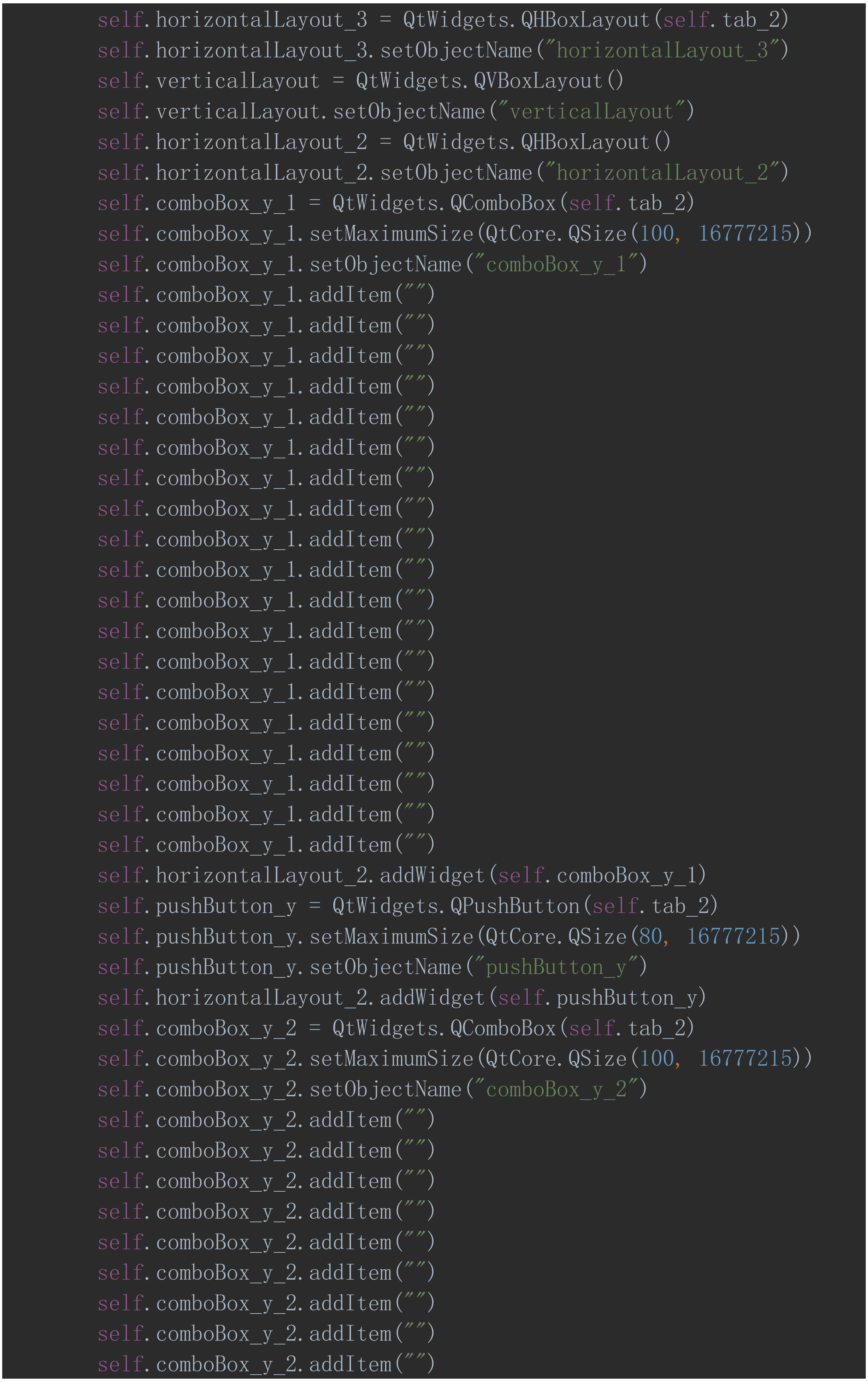
Once PyQt5 is installed, you can create a new window using the QWidget class. To add UI elements to your window, you can use the various layout classes provided by PyQt5, such as QVBoxLayout and QHBoxLayout.

This code creates a window with a QLabel for the input text, a QLineEdit for the user to enter the text to be translated, a QPushButton to initiate the translation, and another QLabel and QLineEdit to display the translated text. The UI elements are added to a QVBoxLayout, which is then set as the layout for the window. The window is then displayed on the screen.

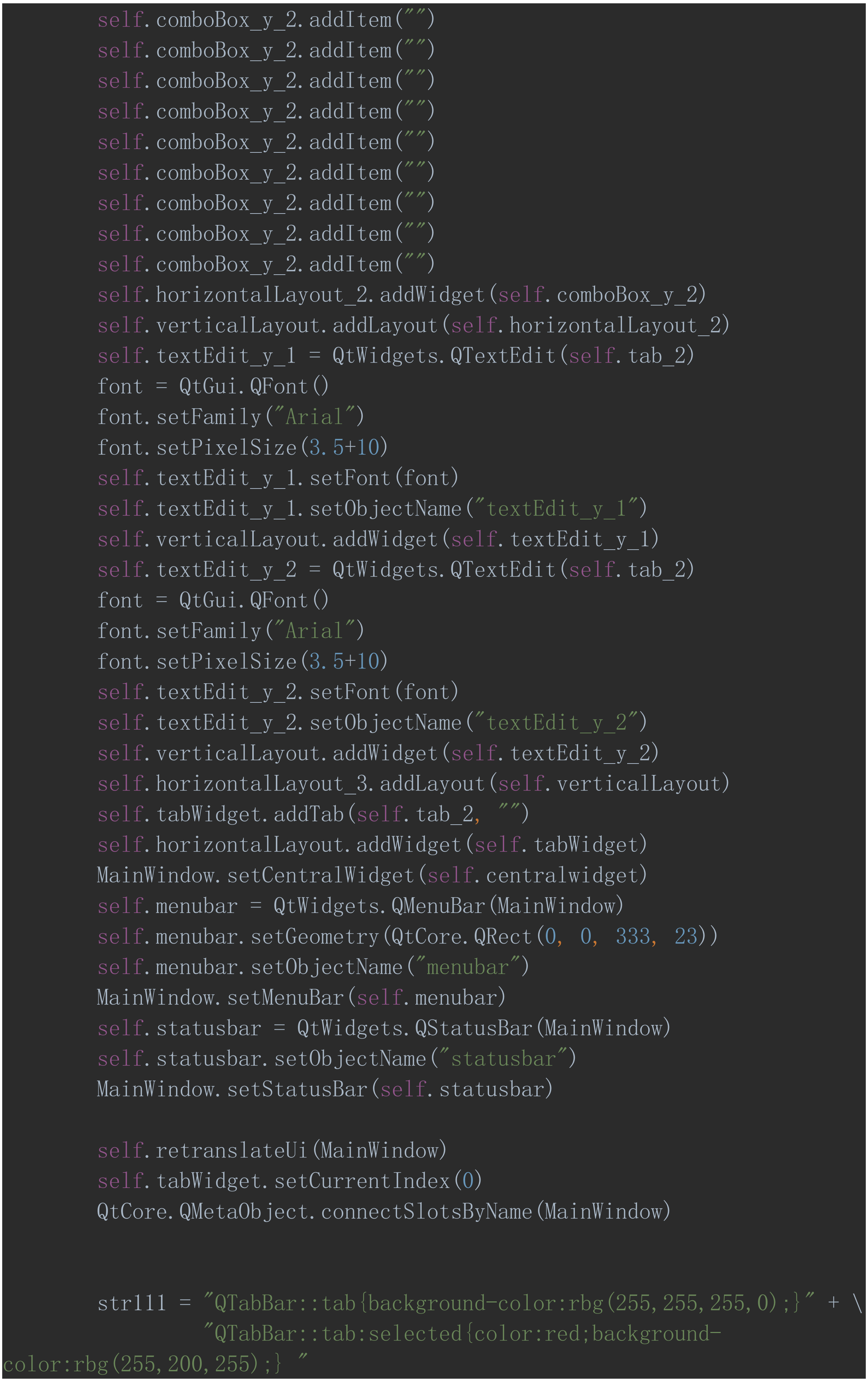
A class called Ui_MainWindow is defined in the window, which inherits from a built-in object class. This class defines a setupUi() method, which is used to initialize a window named MainWindow. The code in the method is mainly used to set various properties of the window, including the size and font of the window, and also defines various controls contained in the window, such as QComboBox, QPushButton, QTextEdit, etc. The code also customizes the appearance and behavior of the window by setting the properties of each control.



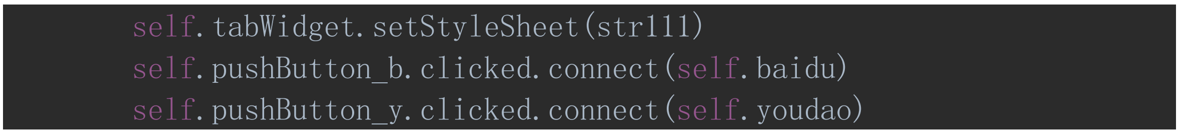
```python
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(333, 440)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.horizontalLayout = QtWidgets.QHBoxLayout(self.centralwidget)
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.tabWidget = QtWidgets.QTabWidget(self.centralwidget)
        font = QtGui.QFont()
        font.setFamily("黑体")
        font.setPixelSize(3.5+12)
        self.tabWidget.setFont(font)
        self.tabWidget.setTabShape(QtWidgets.QTabWidget.Triangular)
        self.tabWidget.setIconSize(QtCore.QSize(20, 16))
        self.tabWidget.setObjectName("tabWidget")
        self.tab = QtWidgets.QWidget()
        self.tab.setObjectName("tab")
        self.horizontalLayout_5 = QtWidgets.QHBoxLayout(self.tab)
        self.horizontalLayout_5.setObjectName("horizontalLayout_5")
        self.verticalLayout_2 = QtWidgets.QVBoxLayout()
        self.verticalLayout_2.setObjectName("verticalLayout_2")
        self.horizontalLayout_4 = QtWidgets.QHBoxLayout()
        self.horizontalLayout_4.setObjectName("horizontalLayout_4")
        self.comboBox_b_1 = QtWidgets.QComboBox(self.tab)
        self.comboBox_b_1.setMaximumSize(QtCore.QSize(100, 16777215))
        self.comboBox_b_1.setObjectName("comboBox_b_1")
        self.comboBox_b_1.addItem("")
        self.comboBox_b_1.addItem("")
        self.comboBox_b_1.addItem("")
        self.comboBox_b_1.addItem("")
        self.comboBox_b_1.addItem("")
        self.comboBox_b_1.addItem("")
        self.comboBox_b_1.addItem("")
        self.horizontalLayout_4.addWidget(self.comboBox_b_1)
        self.pushButton_b = QtWidgets.QPushButton(self.tab)
        self.pushButton_b.setMaximumSize(QtCore.QSize(80, 16777215))
        self.pushButton_b.setObjectName("pushButton_b")
        self.horizontalLayout_4.addWidget(self.pushButton_b)
        self.comboBox_b_2 = QtWidgets.QComboBox(self.tab)
        self.comboBox_b_2.setMaximumSize(QtCore.QSize(100, 16777215))
        self.comboBox_b_2.setObjectName("comboBox_b_2")
        self.comboBox_b_2.addItem("")
        self.comboBox_b_2.addItem("")
        self.comboBox_b_2.addItem("")
        self.comboBox_b_2.addItem("")
        self.comboBox_b_2.addItem("")
        self.comboBox_b_2.addItem("")
        self.comboBox_b_2.addItem("")
        self.horizontalLayout_4.addWidget(self.comboBox_b_2)
        self.verticalLayout_2.addLayout(self.horizontalLayout_4)
        self.textEdit_b_1 = QtWidgets.QTextEdit(self.tab)
        font = QtGui.QFont()
        font.setFamily("Arial")
        font.setPixelSize(3.5+10)
        self.textEdit_b_1.setFont(font)
        self.textEdit_b_1.setObjectName("textEdit_b_1")
        self.verticalLayout_2.addWidget(self.textEdit_b_1)
        self.textEdit_b_2 = QtWidgets.QTextEdit(self.tab)
        font = QtGui.QFont()
        font.setFamily("Arial")
        font.setPixelSize(3.5+10)
        self.textEdit_b_2.setFont(font)
        self.textEdit_b_2.setObjectName("textEdit_b_2")
        self.verticalLayout_2.addWidget(self.textEdit_b_2)
        self.horizontalLayout_5.addLayout(self.verticalLayout_2)
        self.tabWidget.addTab(self.tab, "")
        self.tab_2 = QtWidgets.QWidget()
        self.tab_2.setObjectName("tab_2")
```



```
        self.horizontalLayout_3 = QtWidgets.QHBoxLayout(self.tab_2)
        self.horizontalLayout_3.setObjectName("horizontalLayout_3")
        self.verticalLayout = QtWidgets.QVBoxLayout()
        self.verticalLayout.setObjectName("verticalLayout")
        self.horizontalLayout_2 = QtWidgets.QHBoxLayout()
        self.horizontalLayout_2.setObjectName("horizontalLayout_2")
        self.comboBox_y_1 = QtWidgets.QComboBox(self.tab_2)
        self.comboBox_y_1.setMaximumSize(QtCore.QSize(100, 16777215))
        self.comboBox_y_1.setObjectName("comboBox_y_1")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.comboBox_y_1.addItem("")
        self.horizontalLayout_2.addWidget(self.comboBox_y_1)
        self.pushButton_y = QtWidgets.QPushButton(self.tab_2)
        self.pushButton_y.setMaximumSize(QtCore.QSize(80, 16777215))
        self.pushButton_y.setObjectName("pushButton_y")
        self.horizontalLayout_2.addWidget(self.pushButton_y)
        self.comboBox_y_2 = QtWidgets.QComboBox(self.tab_2)
        self.comboBox_y_2.setMaximumSize(QtCore.QSize(100, 16777215))
        self.comboBox_y_2.setObjectName("comboBox_y_2")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
```



```
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.comboBox_y_2.addItem("")
        self.horizontalLayout_2.addWidget(self.comboBox_y_2)
        self.verticalLayout.addLayout(self.horizontalLayout_2)
        self.textEdit_y_1 = QtWidgets.QTextEdit(self.tab_2)
        font = QtGui.QFont()
        font.setFamily("Arial")
        font.setPixelSize(3.5+10)
        self.textEdit_y_1.setFont(font)
        self.textEdit_y_1.setObjectName("textEdit_y_1")
        self.verticalLayout.addWidget(self.textEdit_y_1)
        self.textEdit_y_2 = QtWidgets.QTextEdit(self.tab_2)
        font = QtGui.QFont()
        font.setFamily("Arial")
        font.setPixelSize(3.5+10)
        self.textEdit_y_2.setFont(font)
        self.textEdit_y_2.setObjectName("textEdit_y_2")
        self.verticalLayout.addWidget(self.textEdit_y_2)
        self.horizontalLayout_3.addLayout(self.verticalLayout)
        self.tabWidget.addTab(self.tab_2, "")
        self.horizontalLayout.addWidget(self.tabWidget)
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 333, 23))
        self.menubar.setObjectName("menubar")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)

        self.retranslateUi(MainWindow)
        self.tabWidget.setCurrentIndex(0)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)


        str111 = "QTabBar::tab{background-color:rbg(255,255,255,0);}" + \
                 "QTabBar::tab:selected{color:red;background-
color:rbg(255,200,255);} "
```



```
        self.tabWidget.setStyleSheet(str111)
        self.pushButton_b.clicked.connect(self.baidu)
        self.pushButton_y.clicked.connect(self.youdao)
```

interface function

Then carry out the relevant design of the interface function and call the Youdao dictionary API to write about the interface function of any language --> Chinese,

Write an interface function about any language --> English,

The interface for translating into three languages is written above. If you want to translate from one language to another, you only need to modify the abbreviation below, which can automatically identify the language that needs to be translated and quickly translate

Choose to use python language 3.8, tool pycharm. Form implementation generated from reading ui file 'window_2.ui' Get the translation result by calling Youdao API interface. In this application, we use the requests library to send translation requests and the json library to parse the response content.

The translate function in the application receives the text to be translated and returns the translated result. It uses the address and key of Youdao Dictionary API to construct request parameters, and uses the requests.get method to send the request.

Before sending the request, the application also needs to call the generate_signature function to generate the Youdao dictionary API signature. This is required by Youdao Dictionary API to verify whether the request comes from an authorized user.

This code implements a simple translation function. It uses Baidu Translate API to translate a piece of text from one language to another.

First, it defines a private method baidu(), which includes all the steps of Baidu translation API calls.

In the method, the first try-except block is used to catch possible exceptions and handle them.

Then, define the URL of Baidu translation API and the appid and secretKey used to call the API, which are provided by Baidu.

Next, a language dictionary Lang is defined to map the names of languages to the abbreviations used in the Baidu Translate API.

Then, get the language and target language of the text that the user wants to translate through the comboBox_b_1 and comboBox_b_2 components, and map them to the abbreviations used by the Baidu translation API.

Next, a random integer is generated as the salt parameter and the value of the sign parameter is calculated.

After that, splice out the complete API request URL, and use the HTTP client to initiate a GET request.

After receiving the response, parse the response content into JSON format, get the translation result and display the result to the user through the textEdit_b_2 component.

where retranslateUi is a function to update the translation of the user interface. In this function, the _translate variable in the first line of code is a Qt translator for translating text into a specific language.

In the second line, the program uses the _translate variable to translate the title text of MainWindow, setting it to the "translation wizard".

Next, the program updates the text in the comboBox_b_1 object by calling the setItemText method of the object, translating them into some languages such as Chinese, English, and Japanese

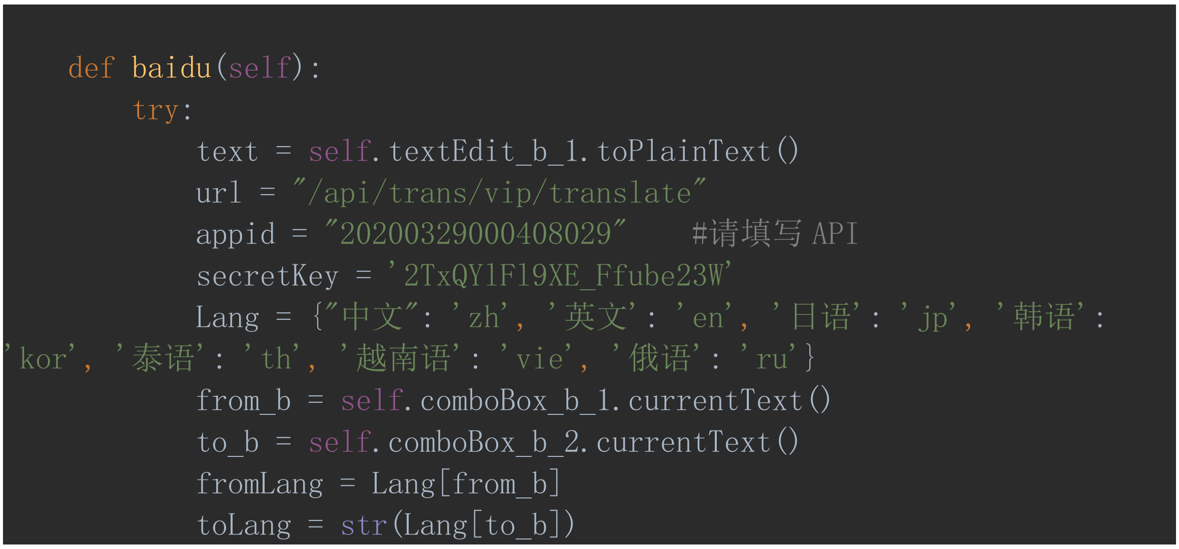
The last if __name__ == '__main__': in this code is a conditional statement. It is used to check whether the code in the code block is the main program, that is, whether it is directly executed.

If the code in the code block is to be run directly, then the value of __name__ will be '__main__'. In this way, the code in the conditional judgment statement will be executed.

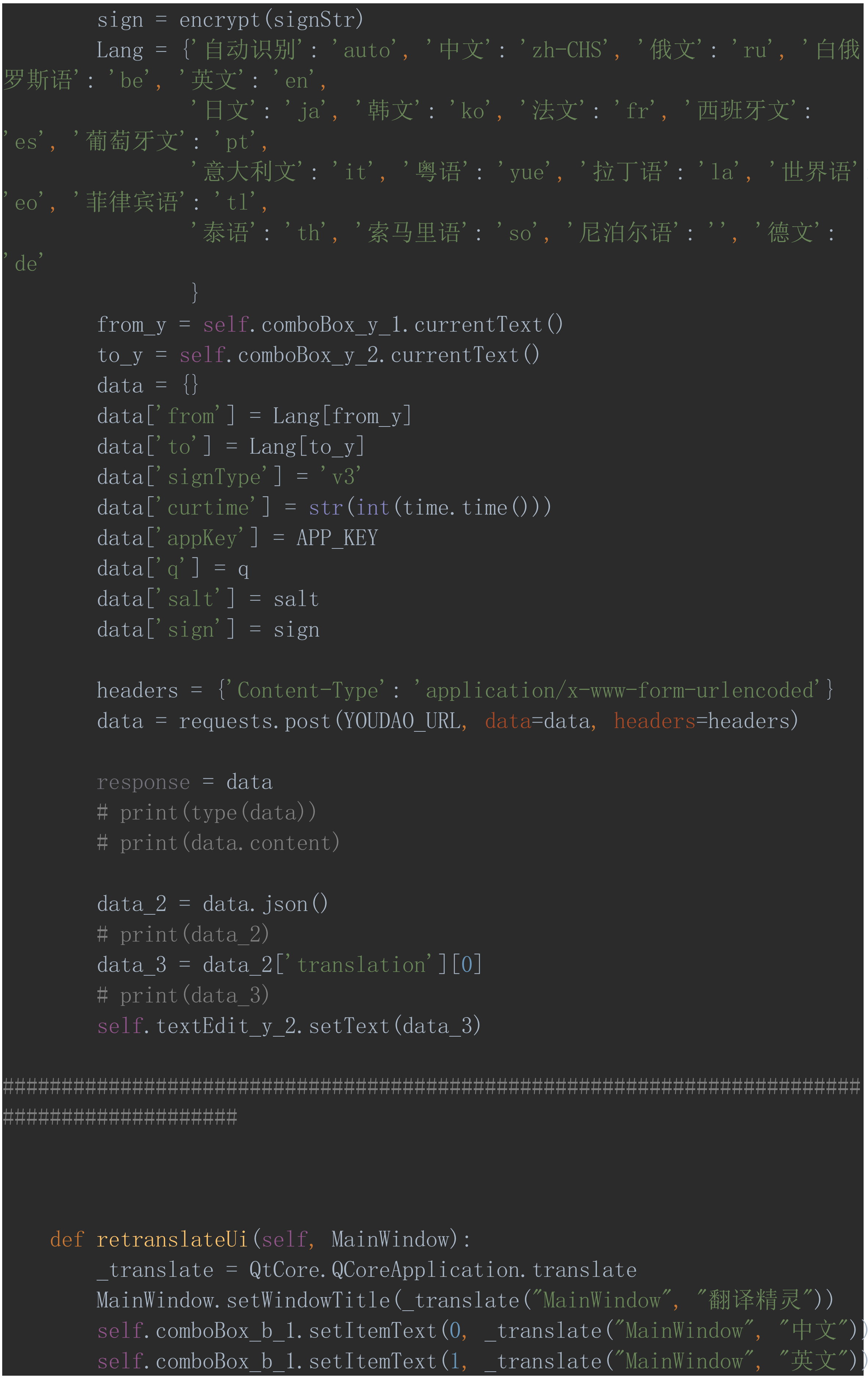
In this code, the part after the if __name__ == '__main__': line of code is the main function of the program. It creates a QApplication object, a QMainWindow object, and a Ui_MainWindow object, and then calls the setupUi() method of the Ui_MainWindow object to set the window interface as the main window. Finally, it calls the show() method of the QMainWindow object to display the window, and calls the exec_() method of the QApplication object to run the program. When a request is sent successfully and a response is returned, the application checks the response status. If the response status is not 200, it means the request failed and the application throws an exception.

If the response status is 200, it means the request is successful, the application will parse the response content, get the translation result, and return the result to the caller. You need to replace <your_api_key> and <your_api_secret> with Youdao Dictionary API key and secret to use this application.

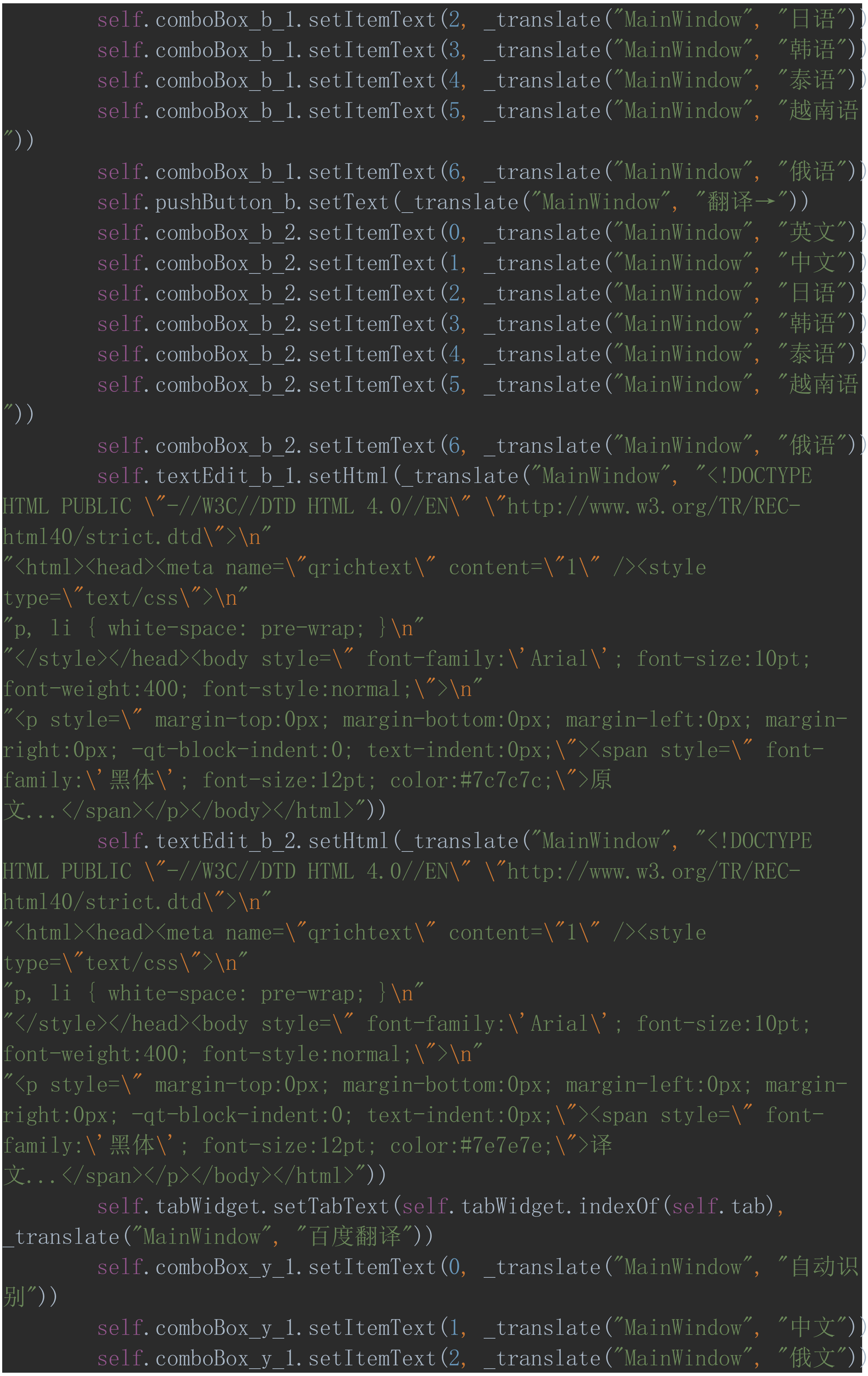
code show as below



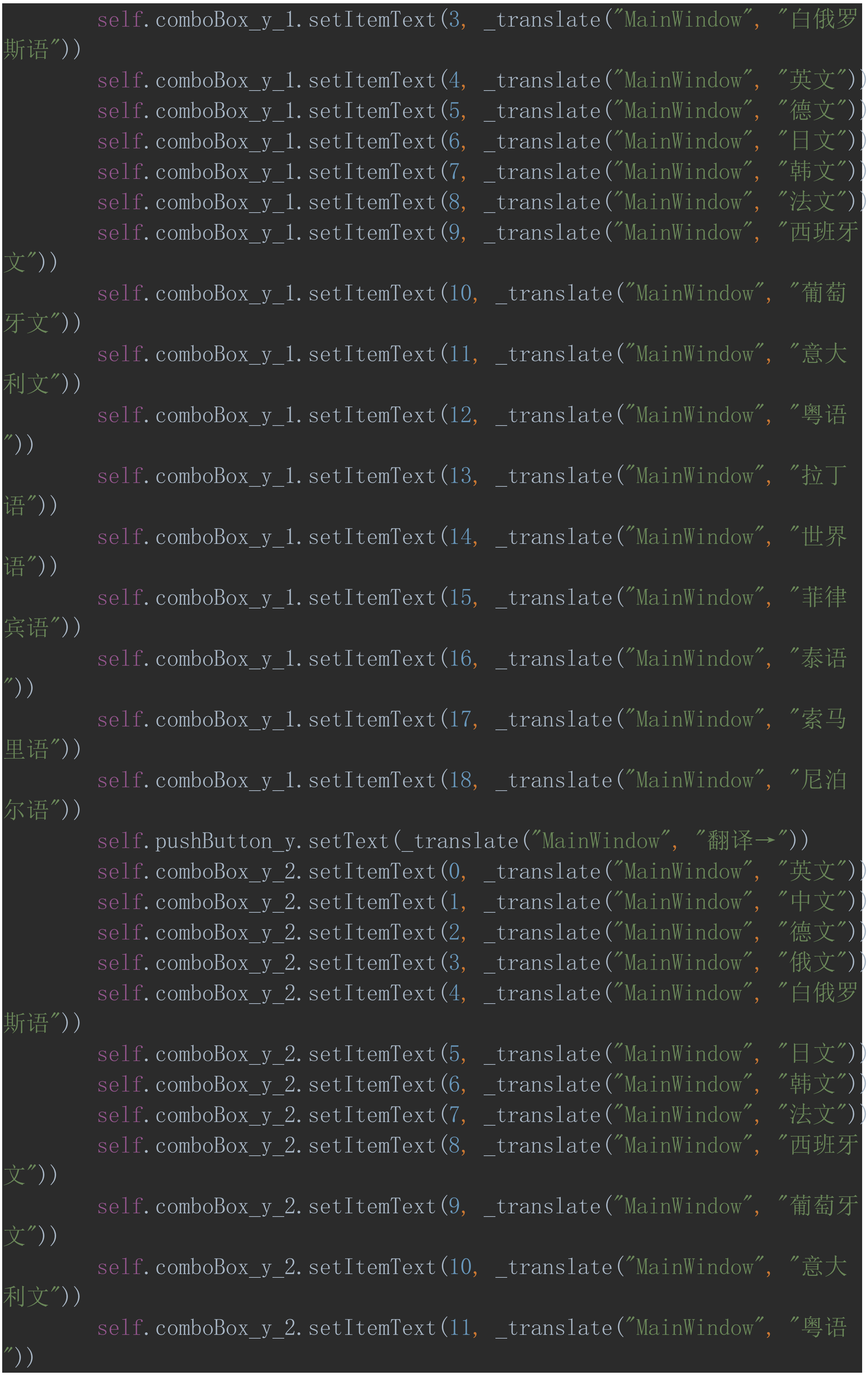
```
    def baidu(self):
        try:
            text = self.textEdit_b_1.toPlainText()
            url = "/api/trans/vip/translate"
            appid = "20200329000408029"    #请填写 API
            secretKey = '2TxQYlF19XE_Ffube23W'
            Lang = {"中文": 'zh', '英文': 'en', '日语': 'jp', '韩语':
'kor', '泰语': 'th', '越南语': 'vie', '俄语': 'ru'}
            from_b = self.comboBox_b_1.currentText()
            to_b = self.comboBox_b_2.currentText()
            fromLang = Lang[from_b]
            toLang = str(Lang[to_b])
            salt = random.randint(32768, 65536)
            sign = appid + text + str(salt) + secretKey
            md = hashlib.md5()
            md.update(sign.encode(encoding='utf-8'))
            sign = md.hexdigest()
            myurl = url + \
                    '?appid=' + appid + \
                    '&q=' + parse.quote(text) + \
                    '&from=' + fromLang + \
                    '&to=' + toLang + \
                    '&salt=' + str(salt) + \
                    '&sign=' + sign
            httpClient = http.client.HTTPConnection('api.fanyi.baidu.com')
            httpClient.request('GET', myurl)
            response = httpClient.getresponse()
            html = response.read().decode('utf-8')
            html = json.loads(html)
            dst = html["trans_result"][0]["dst"]
            self.textEdit_b_2.setText(dst)
        except:
            QMessageBox.warning(self.pushButton, '警告', '错误',
QMessageBox.Yes, QMessageBox.Yes)

    def youdao(self):
        def encrypt(signStr):
            hash_algorithm = hashlib.sha256()
            hash_algorithm.update(signStr.encode('utf-8'))
            return hash_algorithm.hexdigest()

        reload(sys)
        q = self.textEdit_y_1.toPlainText()
        if q is None:
            q_2 = None
        size = len(q)
        if size <= 20:
            q_2 = q
        else:
            q_2 = q[0:10] + str(size) + q[size - 10:size]

        YOUDAO_URL = 'https://openapi.youdao.com/api'
        APP_KEY = '7c62856eed0463b8'
        APP_SECRET = '3HYeqG5ers57Ffqb7VqP5X5Cpw39Xw9B'      #请填写
          salt = str(uuid.uuid1())
        signStr = APP_KEY + q_2 + salt + str(int(time.time())) +
APP_SECRET
```



```
    sign = encrypt(signStr)
    Lang = {'自动识别': 'auto', '中文': 'zh-CHS', '俄文': 'ru', '白俄
罗斯语': 'be', '英文': 'en',
            '日文': 'ja', '韩文': 'ko', '法文': 'fr', '西班牙文':
'es', '葡萄牙文': 'pt',
            '意大利文': 'it', '粤语': 'yue', '拉丁语': 'la', '世界语':
'eo', '菲律宾语': 'tl',
            '泰语': 'th', '索马里语': 'so', '尼泊尔语': '', '德文':
'de'
            }
    from_y = self.comboBox_y_1.currentText()
    to_y = self.comboBox_y_2.currentText()
    data = {}
    data['from'] = Lang[from_y]
    data['to'] = Lang[to_y]
    data['signType'] = 'v3'
    data['curtime'] = str(int(time.time()))
    data['appKey'] = APP_KEY
    data['q'] = q
    data['salt'] = salt
    data['sign'] = sign

    headers = {'Content-Type': 'application/x-www-form-urlencoded'}
    data = requests.post(YOUDAO_URL, data=data, headers=headers)

    response = data
    # print(type(data))
    # print(data.content)

    data_2 = data.json()
    # print(data_2)
    data_3 = data_2['translation'][0]
    # print(data_3)
    self.textEdit_y_2.setText(data_3)

##########################################################################
####################



  def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "翻译精灵"))
    self.comboBox_b_1.setItemText(0, _translate("MainWindow", "中文"))
    self.comboBox_b_1.setItemText(1, _translate("MainWindow", "英文"))
```



```
        self.comboBox_b_1.setItemText(2, _translate("MainWindow", "日语"))
        self.comboBox_b_1.setItemText(3, _translate("MainWindow", "韩语"))
        self.comboBox_b_1.setItemText(4, _translate("MainWindow", "泰语"))
        self.comboBox_b_1.setItemText(5, _translate("MainWindow", "越南语
"))
        self.comboBox_b_1.setItemText(6, _translate("MainWindow", "俄语"))
        self.pushButton_b.setText(_translate("MainWindow", "翻译→"))
        self.comboBox_b_2.setItemText(0, _translate("MainWindow", "英文"))
        self.comboBox_b_2.setItemText(1, _translate("MainWindow", "中文"))
        self.comboBox_b_2.setItemText(2, _translate("MainWindow", "日语"))
        self.comboBox_b_2.setItemText(3, _translate("MainWindow", "韩语"))
        self.comboBox_b_2.setItemText(4, _translate("MainWindow", "泰语"))
        self.comboBox_b_2.setItemText(5, _translate("MainWindow", "越南语
"))
        self.comboBox_b_2.setItemText(6, _translate("MainWindow", "俄语"))
        self.textEdit_b_1.setHtml(_translate("MainWindow", "<!DOCTYPE
HTML PUBLIC \"-//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-
html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style
type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:\'Arial\'; font-size:10pt;
font-weight:400; font-style:normal;\">\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\"><span style=\" font-
family:\'黑体\'; font-size:12pt; color:#7c7c7c;\">原
文...</span></p></body></html>"))
        self.textEdit_b_2.setHtml(_translate("MainWindow", "<!DOCTYPE
HTML PUBLIC \"-//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-
html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style
type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:\'Arial\'; font-size:10pt;
font-weight:400; font-style:normal;\">\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\"><span style=\" font-
family:\'黑体\'; font-size:12pt; color:#7e7e7e;\">译
文...</span></p></body></html>"))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab),
_translate("MainWindow", "百度翻译"))
        self.comboBox_y_1.setItemText(0, _translate("MainWindow", "自动识
别"))
        self.comboBox_y_1.setItemText(1, _translate("MainWindow", "中文"))
        self.comboBox_y_1.setItemText(2, _translate("MainWindow", "俄文"))
```



```
        self.comboBox_y_1.setItemText(3, _translate("MainWindow", "白俄罗斯语"))
        self.comboBox_y_1.setItemText(4, _translate("MainWindow", "英文"))
        self.comboBox_y_1.setItemText(5, _translate("MainWindow", "德文"))
        self.comboBox_y_1.setItemText(6, _translate("MainWindow", "日文"))
        self.comboBox_y_1.setItemText(7, _translate("MainWindow", "韩文"))
        self.comboBox_y_1.setItemText(8, _translate("MainWindow", "法文"))
        self.comboBox_y_1.setItemText(9, _translate("MainWindow", "西班牙文"))
        self.comboBox_y_1.setItemText(10, _translate("MainWindow", "葡萄牙文"))
        self.comboBox_y_1.setItemText(11, _translate("MainWindow", "意大利文"))
        self.comboBox_y_1.setItemText(12, _translate("MainWindow", "粤语"))
        self.comboBox_y_1.setItemText(13, _translate("MainWindow", "拉丁语"))
        self.comboBox_y_1.setItemText(14, _translate("MainWindow", "世界语"))
        self.comboBox_y_1.setItemText(15, _translate("MainWindow", "菲律宾语"))
        self.comboBox_y_1.setItemText(16, _translate("MainWindow", "泰语"))
        self.comboBox_y_1.setItemText(17, _translate("MainWindow", "索马里语"))
        self.comboBox_y_1.setItemText(18, _translate("MainWindow", "尼泊尔语"))
        self.pushButton_y.setText(_translate("MainWindow", "翻译→"))
        self.comboBox_y_2.setItemText(0, _translate("MainWindow", "英文"))
        self.comboBox_y_2.setItemText(1, _translate("MainWindow", "中文"))
        self.comboBox_y_2.setItemText(2, _translate("MainWindow", "德文"))
        self.comboBox_y_2.setItemText(3, _translate("MainWindow", "俄文"))
        self.comboBox_y_2.setItemText(4, _translate("MainWindow", "白俄罗斯语"))
        self.comboBox_y_2.setItemText(5, _translate("MainWindow", "日文"))
        self.comboBox_y_2.setItemText(6, _translate("MainWindow", "韩文"))
        self.comboBox_y_2.setItemText(7, _translate("MainWindow", "法文"))
        self.comboBox_y_2.setItemText(8, _translate("MainWindow", "西班牙文"))
        self.comboBox_y_2.setItemText(9, _translate("MainWindow", "葡萄牙文"))
        self.comboBox_y_2.setItemText(10, _translate("MainWindow", "意大利文"))
        self.comboBox_y_2.setItemText(11, _translate("MainWindow", "粤语"))
        self.comboBox_y_2.setItemText(12, _translate("MainWindow", "拉丁语"))
        self.comboBox_y_2.setItemText(13, _translate("MainWindow", "世界语"))
        self.comboBox_y_2.setItemText(14, _translate("MainWindow", "菲律宾语"))
        self.comboBox_y_2.setItemText(15, _translate("MainWindow", "泰语"))
        self.comboBox_y_2.setItemText(16, _translate("MainWindow", "索马里语"))
        self.comboBox_y_2.setItemText(17, _translate("MainWindow", "尼泊尔语"))
        self.textEdit_y_1.setHtml(_translate("MainWindow", "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:\'Arial\'; font-size:10pt; font-weight:400; font-style:normal;\">\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-indent:0; text-indent:0px;\"><span style=\" font-family:\'黑体\'; font-size:12pt; color:#7c7c7c;\">原文...</span></p></body></html>"))
        self.textEdit_y_2.setHtml(_translate("MainWindow", "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:\'Arial\'; font-size:10pt; font-weight:400; font-style:normal;\">\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-indent:0; text-indent:0px;\"><span style=\" font-family:\'黑体\'; font-size:12pt; color:#7e7e7e;\">译文...</span></p></body></html>"))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_2), _translate("MainWindow", "有道翻译"))


if __name__ == '__main__':
  app = QApplication(sys.argv)
  MainWindow = QMainWindow()
  ui = Ui_MainWindow()
  ui.setupUi(MainWindow)
```

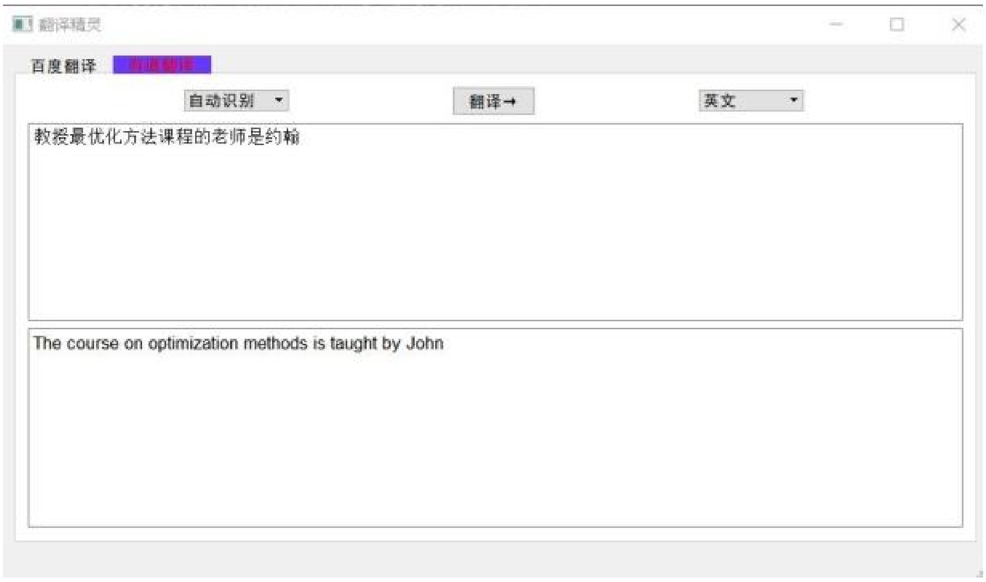
```
    MainWindow.show()
    sys.exit(app.exec_())
```

Chapter 4: **Testing and results**

4.1 Test results for the sign-in feature

We run several sets of related tests,

First, input Chinese "The teacher of the optimization method course is John" for detection, and select the output language as English for detection.

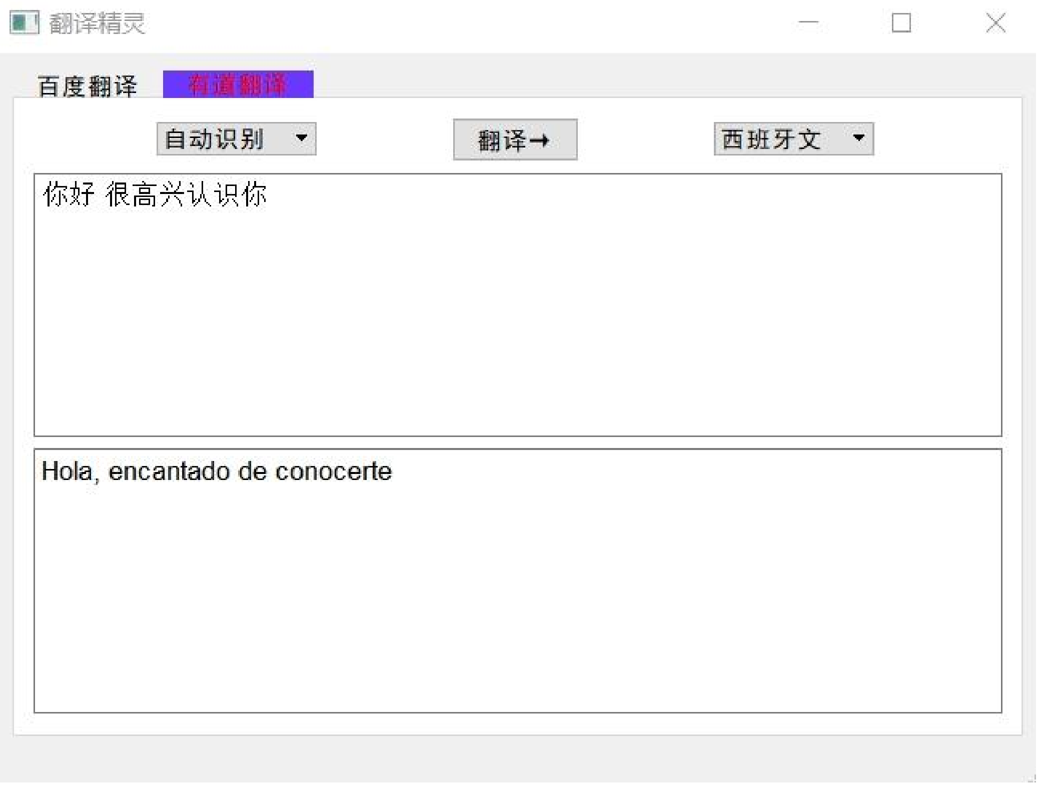


教授最优化方法课程的老师是约翰

The course on optimization methods is taught by John

The results show that the translation is clear, accurate and fast

Then we input "Hi, nice to meet you", select Spanish as the output language for detection.



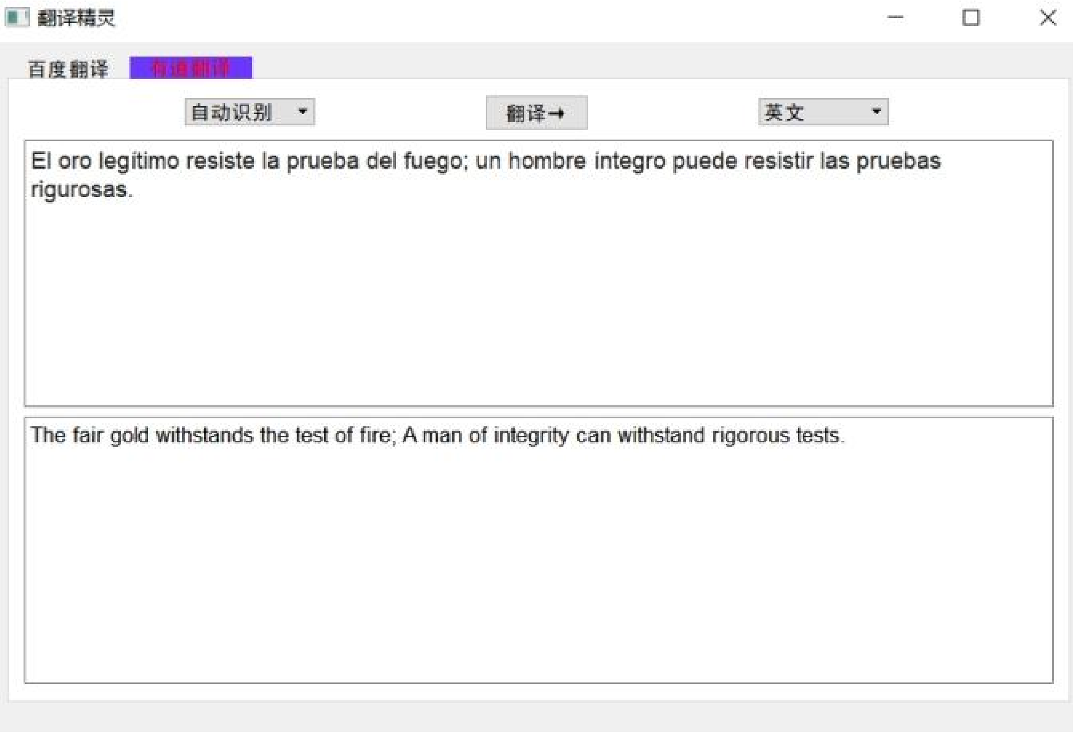
自动识别 西班牙文

你好 很高兴认识你

Hola, encantado de conocerte

The results also show that the translation is clear, accurate and fast

Then we translate the Spanish language to English, check whether it can be automatically recognized, and can be translated into English



自动识别 英文

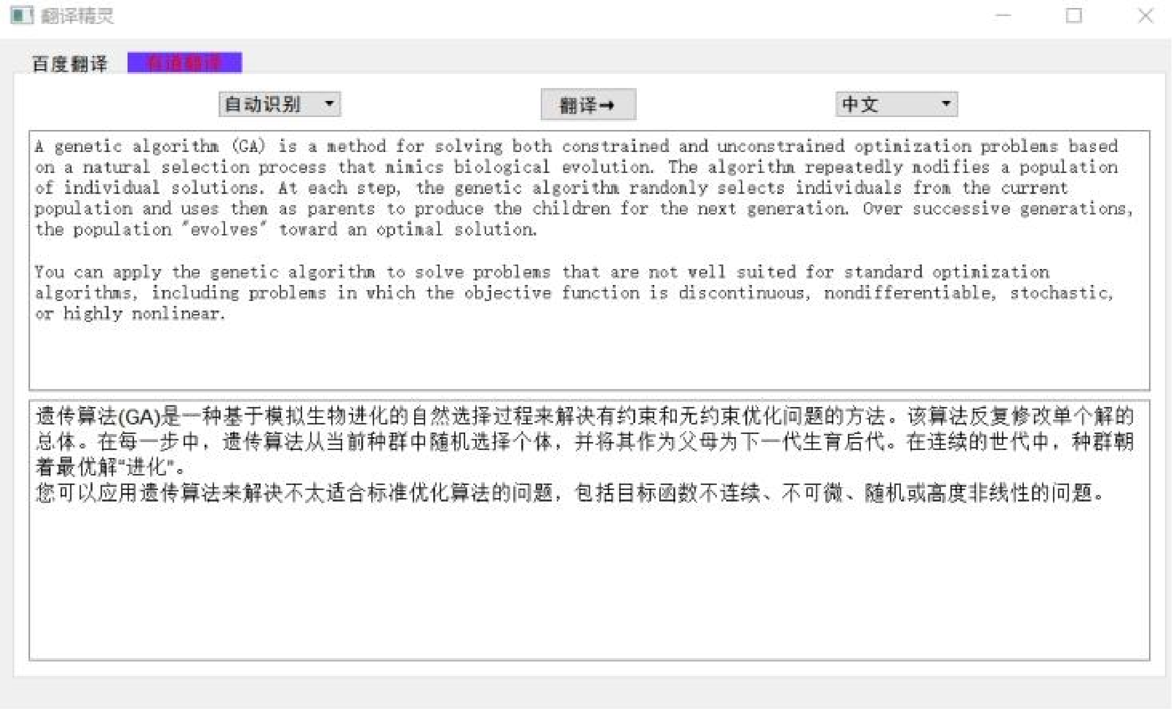
El oro legítimo resiste la prueba del fuego; un hombre íntegro puede resistir las pruebas rigurosas.

The fair gold withstands the test of fire; A man of integrity can withstand rigorous tests.

The results show that the input Spanish can be automatically recognized and translated into the desired output language, The results also show that the translation is clear, accurate and fast

Next we can test long text detection. A genetic algorithm (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The algorithm repeatedly modifies a population of individual solutions. At each step , the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution.

You can apply the genetic algorithm to solve problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear. Enter an English test to translate it into Chinese, and we enter English genetic knowledge of the algorithm and then test it



A genetic algorithm (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution.

You can apply the genetic algorithm to solve problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear.

遗传算法(GA)是一种基于模拟生物进化的自然选择过程来解决有约束和无约束优化问题的方法。该算法反复修改单个解的总体。在每一步中，遗传算法从当前种群中随机选择个体，并将其作为父母为下一代生育后代。在连续的世代中，种群朝着最优解"进化"。
您可以应用遗传算法来解决不太适合标准优化算法的问题，包括目标函数不连续、不可微、随机或高度非线性的问题。

It was found that the software can automatically recognize the language and output it as the specified related language, and it also has certain stability in long texts, and the recognition and translation are fast and accurate

4.2Advantages and disadvantages analysis

The beauty of translation software is that it can automatically convert from one language to another so that people don't need to be multilingual to be able to communicate with people around the world. It can also provide fast, accurate translations, which is especially useful for those who need urgent translations.

The downside of translation software is that it cannot translate all content with complete accuracy, especially content that contains cultural differences, obscure or colloquial language. Sometimes translation software translates sentences into meaningless content, requiring human intervention to improve translation quality. In addition, since translation software relies on human-trained models, its translation quality may also be affected by language development and changes.

The beauty of translation software is that it can automatically convert from one language to another so that people don't need to be multilingual to be able to communicate with people around the world. It can also provide fast, accurate translations, which is especially useful for those who need urgent translations. The advantages of translation software also include:

A large amount of content can be translated, and the translation speed is fast.Provides translations in multiple languages, allowing users to easily switch between languages.Ability to recognize and translate content across different genres, domains and linguistic styles.Translation quality can be automatically updated and improved to provide more accurate translations.Translation results can be exported directly to various applications such as documents, email or social media.In conclusion, the beauty of translation software is that it can automatically and quickly translate one language into another, and it can provide high-quality translation results. Apart from the advantages mentioned above, translation software has many other advantages such as:

A wide variety of different types of text can be translated, including theses, news articles, books, website content, and more.Voice content can be translated to facilitate users to communicate through voice.A variety of different translation modes can be provided, such as offline translation, online translation, instant translation, etc.Various input methods can be provided, such as inputting text, voice or image, etc.Provide real-time translation results to facilitate users to understand the translation progress in real time.

In short, translation software has many advantages, can meet various needs of users for translation functions, and can provide high-quality translation results.

Challenges and Limitations of Machine Translation:

Although Youdao Dictionary translation API provides high-quality text translation, machine translation still has some challenges and limitations.

One challenge is the accuracy of text translation. Although the translation provided by Youdao Dictionary Translation API is accurate, but it is not guaranteed to be completely accurate. Machine translation cannot understand the meaning and context of language, so it can sometimes produce less accurate translations.

Another challenge is the complexity of the language. Some languages, such as Chinese, have more complex grammatical and script features than others, so more problems may arise when translating these languages.

In addition, machine translation also has some limitations. For example, it can only translate text, so it cannot translate speech or images. Also, it cannot translate certain texts, such as those in slang and colloquial speech.

Ways to deal with these problems:

While machine translation presents some challenges and limitations, there are ways to address them.

First, developers can improve the accuracy of text translations by providing multiple versions of the translation. For example, an application could use multiple translation services and then compare the resulting translations to find the most accurate one.

Second, in the case of translating complex languages, professional translators can be used to check and correct the translation. This ensures that translations are as accurate as possible.

In addition, for texts that cannot be translated, human

Work way to deal with. For example, for text in slang and colloquial languages, the app could provide an option to allow the user to manually enter the translation. This ensures that users get the most accurate translation possible.

Overall, using the Youdao Dictionary Translation API to develop text translation applications can provide useful tools for people in various fields to improve the accuracy and convenience of text translation. Although there are some challenges and limitations, these issues can be addressed with appropriate methods.

4.3improvement

This code uses the Translator class from the googletrans library to translate the given text from the source language to the target language. The user is prompted to enter the text, the source language, and the target language, and then the translated text is printed.

You can also add more features to this translator, such as the ability to select from a list of languages or to automatically detect the source language. You can also add options for the user to customize the translation, such as choosing between different translation models or specifying the context or meaning of the text.

If you want to make the text translator more complicated, you can add more features and options to make it more versatile and powerful. For example, you could add the following features:

Language detection: Instead of asking the user to specify the source language, you can use a language detection algorithm to automatically detect the source language of the text. This can be useful if the user is not sure of the language or if the text contains multiple languages.

Language selection: You can provide a list of supported languages and allow the user to choose the source and target languages from the list. This can make it easier for the user to select the correct languages and avoid errors.

Customization options: You can add options for the user to customize the translation, such as choosing between different translation models, specifying the context or meaning of the text, or adjusting the level of fluency or formality of the translation.

Additional functions: You can add additional functions to the translator, such as the ability to save and load translations, translate entire documents or files, or translate text in real-time as it is typed or spoken.

Here is an example of how you could implement some of these features in the text translator:

you want to continue adding more features and options to the text translator, you can consider implementing the following:

Real-time translation: You can add the ability to translate text in real-time as it is typed or spoken. This can be useful for translating conversation in real-time, such as in a chat or call.

Document and file translation: You can add the ability to translate entire documents or files. This can be useful for translating large amounts of text, such as books, articles, or websites.

Multiple translations: You can add the ability to generate multiple translations of the same text, using different translation models, contexts, or languages. This can be useful for comparing different translations or for finding the best translation for a given text.

User interface: You can improve the user interface of the translator, such as by adding a graphical user interface or by providing clear instructions and options for the user. You can also add error handling and validation to prevent invalid input and to ensure that the translator works correctly.

Here is an example of how you could implement some of these additional features in the text translator:

If you want to continue adding more features and options to the text translator, you can consider implementing the following:

User interface: You can improve the user interface of the translator, such as by adding a graphical user interface or by providing clear instructions and options for the user. You can also add error handling and validation to prevent invalid input and to ensure that the translator works correctly.

Custom translation models: You can allow the user to specify custom translation models, such as by providing their own training data or by using advanced machine learning techniques to improve the accuracy and fluency of the translations.

Language learning: You can add features to the translator to help users learn new languages, such as by providing vocabulary lists, pronunciation guides, and practice exercises.

Integration with other services: You can integrate the text translator with other services, such as cloud storage, translation memory databases, or machine translation APIs, to improve the performance and capabilities of the translator.

Chapter 5: **Conclusion**

In this project, I developed an intelligent translation software using Python programming language and QT library. The software integrates Youdao Dictionary API, can translate 104 languages, and focuses on technical and academic translation. The overall development of the software follows a modular approach and undergoes rigorous testing to ensure its effectiveness. The resulting software is efficient, scalable and suitable for individual translation needs, fulfilling the original project vision and goals

Bibliography

[1] HUANG Hao. Encapsulation and Utilization of API of Baidu Translation and Youdao Translation [J]. Computer Programming Skills & Maintenance, 2014(16):4.

[2] Liu Jinhu. A Method to implement API Hook Testing in Python: 2007.

[3] LU Wenzhou. Qt 5 Development and Examples [M]. Publishing House of Electronics Industry, 2015.

[4] Lakshminarayanan V , Ghalila H , Ammar A , et al. GUI Programming with Python and Qt[M]. 2018.

[5] Feng Zhiwei. Research on Machine Translation [M]. China Translation and Publishing Corporation, 2004.

[6] Wang Jing, Xie Cong. Comparison of computer translation tools based on translation methods——Taking Systran, Google Translate, and Trados as examples [J]. English Square: Academic Research, 2016(7): 3.

[7] Ji Yiwei. Quality Evaluation of Chinese Passive Voice Machine Translation——Taking Google Translate and Youdao Translation as Examples [J]. English Square: Academic Research, 2019(3):3.

[8] Sun S Q . Improving automatic evaluation of machine translation using simple linguistic features[J]. journal of computer science\&\technology, 2011.