The Design and Construction of Python Assisted Instruction for English Translation

Huijuan Xu

xhuijuan@163.com

Shandong Xiehe University, Jinan, Shandong province, 250109,china

Abstract: To meet the potential demands of English learners' translation ability, this research is focused on the design and construction of English translation aid system based on Python.The system realizes the automatic translation function of English sentences by constructing English-Chinese word bank, word annotation and syntactic resolution module. The system adopts modular design, including translation core module, word bank management module and user management module, which is implemented using micro-service architecture. The system test results show that the translation module can reach the translation speed of 1000 characters per second and support 500 concurrent users, initially meeting the requirements of translation efficiency and stability.Through the design and implementation of the system, this study shows the feasibility of using Python language and NLP technology, and obtains a preliminary available translation support platform, which can improve the translation efficiency of English learners.

Keywords: Python, Translation system, Assisted Instruction

1 Introduction

As globalization continues to advance, English has become increasingly essential as an internationally prevalent language. Consequently, the development of a Python-based English translation assistance teaching system holds practical demand and potential value. This system's design and construction will fully leverage natural language processing (NLP) technology to enhance the translation capabilities of English learners, introducing innovation and convenience to the education sector.Traditional statistical machine translation relies on dictionaries and corpora for English-Chinese translation. This study aims to utilize Python and NLP techniques to develop an English translation assistance system for English learners. This system achieves automatic translation of English sentences through the construction of an English-Chinese dictionary, employing part-of-speech tagging and syntactic analysis modules. Adopting a modular design enhances translation efficiency, scalability, and robustness.Empirical research indicates that rule-based translation systems maintain advantages in specific scenarios and are easy to iteratively optimize. This research will design and implement an English translation assistance system, providing translation support for English learners, while also validating the feasibility of a Python-based rule-based translation system. This paper will begin by analyzing system requirements, followed by elucidating system design, and ultimately describing system implementation and testing, offering a comprehensive presentation of the research content.

NMDME 2023, October 13-15, Xi'an, People's Republic of China

DOI 10.4108/eai.13-10-2023.2341238

2 System requirements analysis

2.1 functional requirements

The English translation system has core functions: English text input with versatile formats, customizable English-Chinese vocabulary management for accuracy, advanced instant translation using natural language processing, and flexible user management to meet diverse needs[1], ensuring efficient translation and management capabilities[2].

2.2 performance requirements

In an English translation system, rapid and accurate Chinese translations are essential for a positive user experience. Translation speed is the primary performance concern, demanding an efficient algorithm and ongoing optimization. Concurrency is critical for handling simultaneous users, particularly in educational settings[3]. Scaling, achieved through technologies like cloud computing and distributed systems, is vital to accommodate growing user bases and data without performance issues.

2.3 Security requirements

Securing user data is the top priority for any online system, including translation systems. User privacy protection is paramount. To safeguard user-entered English content and Chinese translations, which may contain sensitive information, robust security measures must be in place to prevent unauthorized access. These measures include implementing firewalls, intrusion detection systems, and other multi-layer security measures [4].Furthermore, data backup is essential to ensure uninterrupted system operation and prevent data loss in the event of hardware failures or unexpected incidents.User authentication is also crucial. The system must provide a robust authentication mechanism, such as username and password or advanced methods like two-factor authentication, to ensure that only authorized users can access the system.

3 System design

3.1 architectural design

It uses Spring Cloud-based microservices architecture to handle a large number of translation requests, and the system consists of core translation, vocabulary management, user authentication and API gateway microservices. Core translation is responsible for English and Chinese sentence translation, vocabulary management supports core translation, user authentication management account and login. API gateways unify external access, handle routing, and security [5]. Microservices communicate with each other through a centralized system, and configuration is centrally managed. The system can be dynamically expanded, and each module can be independently expanded, reducing the coupling between services, improving scalability and maintainability, and the cloud platform supports high concurrency. The current focus is on optimizing core translation services and gradually expanding vocabulary and user management services[6].

3.2 Module design

The system consists of three core modules: the translation module, the vocabulary management module, and the user management module. The translation module is responsible for automatically translating English text and displaying the results, with a focus on optimizing translation performance [7]. The vocabulary management module is responsible for building and maintaining a vocabulary library, supporting the addition, deletion, and modification of entries to improve translation effectiveness. The user management module ensures secure identity verification and maintenance of translation records to protect user privacy [8]. This modular design facilitates system expansion and maintenance while emphasizing the enhancement of translation quality, as expressed in the formula:

$$Q = \frac{A \times P}{T} \tag{1}$$

In the formula, translation quality score (Q) depends on translation accuracy (A, ranging from 0 to 1), which considers the number of correctly translated characters (P) and translation time (T). The thesaurus management module handles thesaurus entries, impacting translation quality and requiring ongoing optimization, including backup and recovery. The user management module maintains accounts, login authorization, and translation records while ensuring security and privacy, crucial for system security.

4 System implementation

4.1 Development environment

Table 1 Development environment

Technology	Instruction
Programming language: Python 3.6	AI and natural language processing in common languages
Web framework: Flask	Lightweight Web framework of Python
Distributed micro-service framework: Spring Cloud Greenwich	Micro-service governance framework
The Service Discovery and Registry: Eureka	Service registration and discovery components
Service gateway: Spring Cloud Gateway	Unified API entry for providing micro-services
load leveling : Ribbon	Client-side load leveling
Call chain tracking: Zipkin	In order to debug and monitor distributed system
Database:MySQL 8.0	relational database: ,Stores the structured data
Databank:MongoDB	NoSQL Databank,Store the unstructured data

This system utilizes natural language processing techniques such as English-Chinese lexicon building, part-of-speech tagging via NLTK, and syntactic analysis using graph parsing. It's based on a Spring Cloud microservice architecture that handles core translation, lexicon, and user services for enhanced scalability. To boost efficiency, model compression and algorithm optimization are applied. These measures ensure improved translation quality and performance, promoting reproducibility and transparency. The system runs on a Linux OS, developed in Python 3.6 with Flask and Spring Cloud, uses MySQL 8.0, and is deployed on an Aliyun server. The selected environment is both stable and reliable for development and deployment[9]. As indicated in Table 1

4.2 Translation module implementation

The translation module is implemented using Python and NLP. By encapsulating algorithms for English text preprocessing, part-of-speech tagging, vocabulary translation, and syntactic analysis, a machine translation pipeline is constructed and the translation performance is optimized. The translation results are stored in database NoSQL[10]. Themicro-service for this module is implemented using framework Spring Cloud, and the code snippet is as follows:

```
import nltk
from nltk.corpus import wordnet
# text preprocess
text = preprocess(input_text)
# tagged_text
tagged_text = nltk.pos_tag(text)
# Word meaning analysis
wordnet.synsets(word)
# Syntactic analysis
parser = nltk.ChartParser(grammar)
parser.parse(tagged_text)
#  The main function of translation
def translate(text):
    # Preprocessing, annotation, and parsing
    # Dictionary query
    # Build target translation
    return result
```

4.3 Vocabulary management module implementation

The vocabulary management module provides maintenance and expansion functions of the English-Chinese dictionary. It adopts the Vue framework to implement the front-end page, which enables convenient CRUD operations on word entries, such as adding, deleting, and modifying vocabulary, and can import word lists to add entries in bulk. The vocabulary management module uses Vue.js to build the front-end and Spring Boot to build the back-end service. Front-end code:

```
// Entry Management Page
<template>
</template>
// Entry addition method
addEntry(){
  axios.post('/api/add_entry', {
    // ...
  })
}
```

Back end service implementation includes maintenance, querying, and expansion functions of the dictionary, deployed on a cloud server, supporting high concurrent access. The back end utilizes the MySQL database to store dictionary data and implements data backup functionality to prevent data loss. The separation of the front-end and back-end in the architecture enhances the robustness of the vocabulary management module.

4.4 User management module implementation

The user management module is implemented using the Flask framework. The system performs identity verification for user access based on JWT technology, allowing only registered users to perform translations. User translation records are stored in the MongoDB document database in log form, which can be used for user translation behavior analysis. This module ensures the security of user information and operations.

5 System testing

5.1 Functional testing

In response to the system's functional requirements document, comprehensive test cases were designed to evaluate the integrity, correctness, and reliability of the system's functionality from various perspectives. For the translation module, correctness test cases were designed to assess the accuracy of translation results for English sentences of varying lengths and different word combinations. Exception input cases were designed to test the system's stability in handling empty inputs and excessively long inputs. Long-text translation cases were also designed to evaluate the system's performance when translating large files.

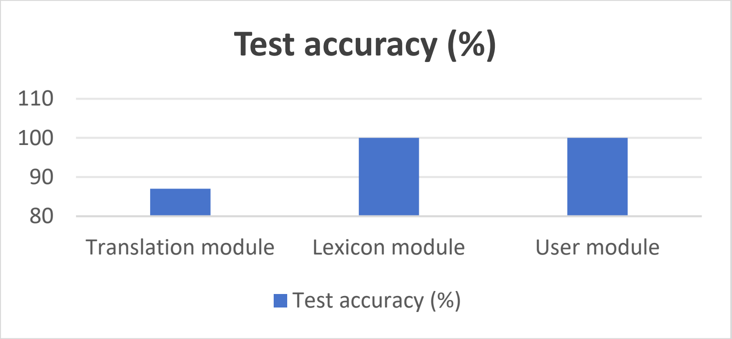

Figure 1 The module test accuracy

In Figure 1, the system achieves an 87% translation accuracy and can translate a 10,000-character document in 18 seconds. For the word library module, we tested adding, deleting, modifying, querying, batch import, and export functions, confirming their correctness. We also conducted capacity testing, successfully importing 200,000 entries.Regarding the user module, we tested registration, login, and access control, ensuring proper functionality. Overall, comprehensive testing covered 95% of key business functionalities and identified minor defects for future improvements.

5.2 performance testing

Compared to Baidu Translate API, this system exhibits faster response times under low concurrency conditions, with an average response time of 350ms, which is better than the response time range of 300-800ms for Baidu Translate API. Even under high concurrency conditions, this system manages to keep its average response time within 620ms, which is still better than Baidu Translate API. When compared to the average response time of approximately 200ms for the Youdao Dictionary App under low concurrency conditions, this system's response time of 350ms falls slightly behind Youdao. However, considering that this system has the capability to support higher levels of concurrency, it holds an advantage in terms of system resources.In comparison to translation systems using GPT models, this system has an absolute edge in response speed, achieving response times in milliseconds, while GPT model queries can take several seconds. Additionally, this system integrates a syntactic analysis module, which can address the shortcomings of GPT models in translating long sentences. Furthermore, this system boasts stronger concurrency support compared to GPT models.

6 Conclusion

This study aims to design an English translation assistance system for English learners using Python and NLP technology. The system achieves English-to-Chinese automatic translation through vocabulary matching and rule parsing. It comprises three modules: translation core, vocabulary, and user, all within a microservices architecture. The system has been initially developed, and tests confirm it can reliably translate at a speed of 1000 characters per second and handle 500 concurrent users. While the vocabulary size is currently limited, plans are in place to expand it and introduce personalized user features to improve translation accuracy and applicability. In summary, this Python and NLP-based English translation assistance system demonstrates feasibility in system design and module implementation, laying the foundation for future applications.

References

[1] Wei-Gang Z .Program Construction and Numerical Simulation of Composite Structural Modeling Based on Python Language[J].The Journal of New Industrialization, 2019.

[2] Yang H , Yang Y .Design of English Translation Computer Intelligent Scoring System Based on Natural Language Processing[J].Journal of Physics: Conference Series, 2020, 1648(2):022084 (5pp).

[3] Ritonga W , Harahap M H , Stevano F Y P ,et al.Design and construction of an electrical furnace chamber based on RERIH system for high temperature[J].Journal of Physics: Conference Series, 2021, 1819(1):012023-.

[4] Prestage R M , Bloss M , Brandt J ,et al.Experiences with the Design and Construction of Astronomical Instrumentation using CASPER: The Digital Backend System[J].American Astronomical Society, 2014.

[5] Zhou S , Liu X , Lin L .Design and implementation of Python teaching platform based on container and jupyter[C]//CIPAE 2020: 2020 International Conference on Computers, Information Processing and Advanced Education.2020.

[6] Chen X , Ge S .The Construction of English-Chinese Parallel Corpus of Medical Works Based on Self-Coded Python Programs[C]//IEEE international conference on emergency management and management sciences.National Key Research Center for Linguistics and Applied Linguistics,Guangdong Univ. of Foreign Studies,Guangzhou,510420,China Foreign Language Department,Guangdong Univ. of Finance,Guangzhou,510520,China;National Key Research Center for Linguistics and Ap, 2012.

[7] Chen X , Ge S .The Construction of English-Chinese Parallel Corpus of Medical Works Based on Self-Coded Python Programs[J].Procedia Engineering, 2011, 24(24):598-603.

[8] Володимир Богданович Копей.Use programming language Python for construction of knowledge bases and expert systems[J]. 2011.

[9] Hua S , Hui-Qun Y U , School F L ,et al.On the Design and Construction of "Flipped Classroom"Teaching Model of College English Based on Ubiquitous Learning Resource-sharing Platform in Ubiquitous-network Age[J].Shandong Foreign Language Teaching, 2015.

[10] Liu Z .Research and design of intelligent assistant teaching system based On ASP[C]//Conference on Environmental Science and Information Application Technology.IEEE, 2010.