Pokhara University

CITIZEN COLLEGE

PROJECT NO.: 2021-1-53-0020 / 2021-1-53-0021

LANGUAGE TRANSLATOR USING PYTHON

BY

BIDHYA BARAM & BIJAY TAMANG

A PROJECT REPORT

SUBMITTED TO THE DEPARTMENT OF BACHELOR OF COMPUTER APPLICATION

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF COMPUTER APPLICATION

CITIZEN COLLEGE, LALITPUR, NEPAL

JULY 28, 2023

Language Translator using Python

by

Bidhya Baram & Bijay Tamang

2021-1-53-0020 / 2021-1-53-0021

Project Supervisor

Er. Nishan Khanal

A project submitted in partial fulfillment of the requirements for the degree of

Bachelor of Computer Application

Department of Bachelor of Computer Application

Pokhara University, Citizen College

Lalitpur, Nepal

July 28, 2023

COPYRIGHT ©

DECLARATION

We declare that the work hereby submitted for Bachelor in Computer Application at the Pokhara University, Citizen College entitled "**Language Translator using Python**" is our own work and has not been previously submitted by us at any university for any academic award. We authorize the Pokhara University, Citizen College to lend this project work to other institutions or individuals for the purpose of scholarly research.

Bidhya Baram & Bijay Tamang
2021-1-53-0020 / 2021-1-53-0021
July 28, 2023

RECOMMENDATION

The undersigned certify that they have read and recommend to the Department of Bachelor of Computer Application for acceptance, a project work entitled "**Language Translator using Python**", submitted by **Bidhya Baram & Bijay Tamang** in partial fulfillment of the requirement for the award of the degree of "**Bachelor of Computer Application**".

Project Supervisor

Er. Nishan Khanal

Lecturer/Researcher

BCA Program Coordinator

Er. Nishan Khanal

Department of Bachelor of Computer Application, Citizen College

July 28, 2023

DEPARTMENTAL ACCEPTANCE

The project work entitled "**Language Translator using Python**", submitted by **Bidhya Baram & Bijay Tamang** in partial fulfillment of the requirement for the award of the degree of "**Bachelor of Computer Application**" has been accepted as a genuine record of work independently carried out by the student in the department.

Head of the Department
Department of Bachelor of Computer Application,
Citizen College,

July 28, 2023

LETTER OF APPROVAL

We certify that we have examined this report entitled"**Language Translator using Python**", submitted by **Bidhya Baram & Bijay Tamang** in and are satisfied with the project defense. In our opinion it is satisfactory in the scope and qualify as project in partial fulfillment of the requirements for the"**Bachelor of Computer Application**" under Department of Bachelor of Computer Application, **Pokhara University**.

Project Supervisor

Er. Nishan Khanal

Lecturer/Researcher

July 28, 2023

Examiner

July 28, 2023

BCA Program Coordinator

Er. Nishan Khanal

Department of Bachelor of Computer Application, Citizen College

July 28, 2023

ACKNOWLEDGMENT

This project work would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First of all, we would like to express our sincere gratitude to our supervisor, **Er. Nishan Khanal** , of **Citizen College** for providing invaluable guidance, insightful comments, meticulous suggestions, and encouragement throughout the duration of this project work. Our sincere thanks also goes to the BCA coordinator, **Er. Nishan Khanal**, for coordinating the project works, providing astute criticism, and having inexhaustible patience.

We are also grateful to our classmates and friends for offering us advice and moral support. To our family, thank you for encouraging us in all of our pursuits and inspiring us to follow our dreams. We are especially grateful to our parents, who supported us emotionally, believed in us and wanted the best for us.

Bidhya Baram & Bijay Tamang
2021-1-53-0020 / 2021-1-53-0021
July 28, 2023

ABSTRACT

The proposed project aims to create a multilingual translator that will allow users to communicate effectively across languages. The application's main objectives include providing a user-friendly interface for language translation and allowing users to contribute translations to enhance the dataset. The GUI interface presents clear instructions and options for translation, making it accessible to users with minimal technical expertise. Users can either use the translation feature or provide their own translations to improve the dataset. To accurately translate content between several languages, the translator will use datasets from csv files for various languages. Users will be able to input and translate material on the translator with ease thanks to its user-friendly interface. As part of the project, translation algorithms will be created and put into use to guarantee user experience and high-quality translation accuracy. A wide spectrum of users, including people, companies, and organizations, will be catered for by the translator. Future enhancements are proposed to address the unfulfilled objectives and improve the application's performance. Integrating advanced language models, such as Neural Machine Translation (NMT) models, can significantly enhance translation accuracy and handle complex sentence structures. Expanding the dataset to include a broader vocabulary and domain-specific terminology would further improve translation quality. Additionally, the application could benefit from real-time translation capabilities and voice input support, making it more dynamic and user-friendly. Integrating external translation APIs would offer access to state-of-the-art language models, while user feedback mechanisms could aid in dataset improvement and translation refinement. The Language Translator, while serving as a basic prototype, requires further development to achieve higher accuracy and functionality. The proposed future enhancements pave the way for an advanced and versatile language translation tool capable of meeting diverse user needs and providing seamless cross-lingual communication.

Keywords: Accuracy,Algorithms,Cross-lingual, Translation,

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	Application Programming Interface
CSV	Comma Separated Values
DFD	Data Flow Diagram
ER	Entity Relationship
ETF	Enhanced Translation Framework
GUI	Graphical User Interface
HOG	Histogram of Oriented Gradients
JSNON	JAVAScript object Notation
NLP	Natural language Processing
OCR	Optical character recognition (OCR)
RST	Rough Set Theory
T2MSTA	Text to Multilingual Speech Translator using Android

1 INTRODUCTION

In today's increasingly globalized world, language translation is a crucial tool for effective communication. However, it can be difficult to translate anything accurately and honestly, especially when using difficult technical or specialist jargon. Our Language Translator project aims to bridge this gap by providing a user-friendly and efficient solution to translate text between multiple languages, all by leveraging the flexibility of CSV files as the underlying data source The proposed project seeks to create a novel language translator. A wide range of users, including people, companies, and organizations from a variety of industries—including healthcare, education, finance, and government would benefit from the proposed language translator, which has been made to suit their needs . Multiple languages has been supported by the translator. A user-friendly and effective language translator has been designed and put into use by the project team's members.Instead of relying on complex databases or external APIs, we've ingeniously chosen to use CSV files to store language translation pairs. This approach makes it easier for users to customize and extend the translations according to their unique needs. We have made use of the CSV files in the project to create a dataset for different languages. Thus, the translation is done through the CSV files. We can also add new words and help improve the further capacity of the language translator.We believe that technology should be accessible to everyone . We have made an easy to use interface, with guidelines and instructions to help the user understand the working of the translator even better. By utilizing Python's efficiency and the lightweight nature of CSV files, our language translator offers blazing-fast translation capabilities, ensuring that you get your translations done in no time . The heart of our project lies in the CSV files, where we maintain the language translation pairs. Each row in the CSV represents a unique translation, with columns specifying the source language, target language, and the corresponding translation. The user interacts with the Python program through graphical user interface (GUI) Ṫhey provide the source text and select the desired source and target languages. The Python program then reads the appropriate CSV file, finds the matching translation pair based on the specified source and target languages, and returns the translated text . The translated text is displayed to the user after translation. The project team members have dedicated to provide users with a high-quality and efficient language translator.

1.1 Background

In today's globalized world, effective communication is paramount for success in various domains, including business, healthcare, and law. However, language barriers often hinder seamless communication, especially in cross-cultural contexts. Language translation plays a crucial role in overcoming these obstacles and facilitating effective communication between individuals speaking different languages [1]. With the rapid advancements in machine learning and natural language processing (NLP) technologies, automated language translation tools have witnessed significant progress, enabling the processing of vast amounts of text and accurate translations across multiple languages [1]. While these advancements have been commendable, current language translation systems still encounter challenges, particularly when dealing with specialized or technical terminology [1]. Such scenarios can significantly impact translation accuracy and reliability, posing potential risks and implications in critical industries where precision is of utmost importance. To address these limitations and enhance language translation services, this project aims to develop a cutting-edge language translator. Leveraging the power of CSV files, this innovative translator seeks to deliver precise and trustworthy translations across multiple languages [1]. The utilization of CSV files allows for the creation of comprehensive datasets containing word pairs for each supported language. This approach enables the translator to perform word-to-word translations with greater accuracy, making it suitable for a wide range of applications [1]. By harnessing the potential of machine learning and NLP, the proposed language translator seeks to provide accurate and context-aware translations , even in domains involving complex technical or specialized language [1]. The incorporation of advanced language models, such as Neural Machine Translation (NMT) systems, would further enhance the translator's capabilities, enabling it to handle intricate sentence structures and contextual nuances [1]. The ultimate goal of this project is to bridge communication gaps and facilitate effective cross-lingual interactions in various sectors, including international business, healthcare, legal proceedings, and academic research [1]. By offering a reliable and versatile language translation tool, this project aims to empower individuals and organizations to communicate seamlessly across linguistic boundaries, fostering collaboration and mutual understanding in our diverse and interconnected world.

1.2 Motivation

The growing demand for efficient cross-lingual communication in today's globalized world serves as the driving force behind the suggested language translator project. Communication problems caused by language hurdles can slow down development in a variety of industries, including healthcare, education, finance, and government. These hurdles can be overcome and effective cross-language and cross-cultural communication facilitated through accurate and trustworthy language translation. Despite major improvements in language translation technology, there are still many problems with current translation systems, especially when working with complicated technical or specialist vocabulary. Inaccurate translation can have serious repercussions, including incorrect diagnoses in the medical field or legal problems in the judicial system. In order to produce precise and trustworthy translations, the proposed language translator will make use of CSV files in Python and use the datasets to do the translations. Multiple languages will be supported by the system. A wide range of people, businesses, and organizations stand to gain from the potential for the proposed language translator to drastically lower language barriers and promote successful communication across languages and cultures. The project team is driven to create a language translator that is high-quality, efficient, and meets user needs.

Figure 1.1: Motivation

1.3 Problem Statement

Language barriers can impede effective communication and hinder progress in various fields, including healthcare, education, finance, and government. While language translation technology has witnessed significant advancements, existing translation systems

still face significant challenges, particularly when dealing with complex technical or specialized terminology. Inaccurate translation can have severe consequences, such as misdiagnosis in healthcare or legal disputes in the justice system. Furthermore, existing language translation systems may not support all languages and may lack advanced features such as speech-to-text and text-to-speech translation. This can limit the effectiveness of cross-lingual communication and hinder progress in various fields. To address these challenges, there is a need for an innovative language translator to provide accurate and reliable translation services. The proposed language translator aims to support multiple languages. The problem statement of this project is to develop a language translator that addresses the challenges of inaccurate translation, limited language support, and lack of advanced features, thereby facilitating effective cross-lingual communication and contributing to progress and development in various fields. The goal of this project is to create a language translator that solves the problems of inaccuracy in translation, a lack of advanced features, and limited language support, enabling efficient cross-lingual communication and advancing various fields.

1.4 Objectives of Project

The proposed language translator project aims to achieve the following objectives:

1. To develop an innovative language translator that utilizes predefined dataset in CSV format to provide accurate and reliable translation services.

2. To support multiple languages and provide advanced features to facilitate cross-lingual communication.

3. To facilitate the feature of adding the translation words into the predefined datasets.

1.5 Scope of Project

The development of a software system that offers precise and dependable translation services for numerous languages will be the main goal of the language translator project. Dataset will be used in the project. The project won't entail creating any hardware parts or tangible objects. Effective cross-linguistic communication is crucial for advancement and growth across a range of industries, including government, finance, healthcare, and education. Inaccurate translation can have serious repercussions, including incorrect

diagnoses in the medical field or legal problems in the judicial system. By addressing the issues of faulty translation and limited language support in current language translation systems, the proposed language translator project intends to promote efficient cross-lingual communication and contribute to advancement and development in numerous fields.The project's intended objective is the creation of an efficient and effective language translator that offers precise and dependable translation services for a variety of languages. The system will have a user-friendly interface and be accessible to people, businesses, and organizations all around the world. The initiative seeks to drastically lower language barriers and enable efficient cross-language and cross-cultural communication for the benefit of people, businesses, and organizations working in a variety of disciplines.

1.6 Potential Applications

A few potential project applications for language translator project are:

Healthcare: The language translation project can be used in contexts where clear communication is essential for providing patients with the best treatment possible. Patients who speak a different language can be communicated with by doctors and nurses using a language translator, ensuring that they obtain appropriate diagnosis and treatments.

Education: The language translator project can be used in educational contexts, allowing teachers to interact with students who speak different languages using the system. Additionally, the technology can be used to translate instructional materials and resources so that a larger audience can access them.

Travel: The language translator project might be used in the tourism sector, allowing travelers to speak with locals in different countries. This can improve the travel experience and make it simpler for visitors to get about strange places.

Business: The language translator project can be applied to businesses operating in multiple countries. The system can be used to facilitate cross-lingual communication between employees, customers, and partners, enabling smoother business operations and expanding market reach.

Government: The language translator project can be applied to government agencies and services to ensure that all citizens have equal access to information and services, regardless of their language. For example, government websites, documents, and services can be translated into multiple languages using the language translator

1.7 Originality of Project

We have come up with the idea to help reduce the language barrier problem between people with this project. The project is completely new idea and no-one has come up with this topic in our college yet. That is how the content, code and the documentation of the project can be considered to have originality. While there are numerous existing language translation tools, many of them have limitations in terms of usefulness, accuracy, and supported languages. For the translation mechanism, we have prepared 5 data sets for five different languages that will help us add and retrieve the texts for different languages. The proposed project also seeks to create a user-friendly interface that is available to people, companies, and organizations all across the world. It prioritizes as well.

1.8 Organisation of Project Report

The material in this project report is organised into seven chapters. Chapter 1 introduces the problem topic this research tries to address, chapter 2 contains the literature review of vital and relevant publications, pointing toward a notable research gap. Chapter 3 describes the methodology for the implementation of this project. Chapter 4 provides an overview of what has been accomplished. Chapter 5 contains some crucial discussions on the used model and methods. Chapter 6 mentions pathways for future research direction for the same problem or in the same domain. Chapter 7 concludes the project shortly, mentioning the accomplishment and comparing it with the main objectives.

2 LITERATURE REVIEW

2.1 LITERATURE REVIEW I

The study ”Language,transalation and culutre” explores the complicated connections between language, translation, and culture are examined in this essay. The importance of translation as a vital tool for communication, cross-cultural exchange, and knowledge dissemination in the diverse world of languages and cultures is emphasized[2]. The writers stress that although there might not be a particular framework for translation, all translators encounter some restrictions and censorship when transmitting meaning because of the cultural and religious standards of their cultures[2]. The paper explores the difficulties and limitations of the translation process. It underlines that a professional translator needs to have a thorough awareness of cultural aspects, viewpoints, and customs in addition to being fluent in both the source and destination languages[2]. To elicit the same emotion that the original author intended, the cultural backdrop of the source material, including its chronological arrangements, explicit meaning, and historical and religious context, must be intentionally taken into account[2]. Translation professionals must be bilingual, bicultural, if not multilingual, as the goal is not just language transfer but also cultural transposition. The paper makes the case that translation is crucial to the spread of popular culture and thus to the globalization of culture. It serves as a link, promoting communication among many languages and cultural groups[2]. In turn, this aids in the formation of a cultural network and promotes cultural globalization by allowing readers to reflect on the cultural background of the original language. The study also emphasizes how the benefits of the original culture’s habits, rituals, and religious values can be better understood and enjoyed globally through translation[2]. The paper notes that cultural variations cause translation challenges that can cause misunderstandings among readers. Every society creates its own culture based on its historical context, current circumstances, and religious convictions; this culture must be recognized and accepted together with its flaws. The report concludes that in a world with many different languages and cultures, translation is an essential instrument for communication and knowledge sharing[2]. It might be difficult for translators to bridge linguistic gaps while taking into account cultural variations. Translation connects various cultures, creating understanding and a respect for the richness of human variation as globalization continues to have an impact on cultures all over the world.

2.2 LITERATURE REVIEW II

The research paper delves into an in-depth analysis of the challenges faced by Saudi Arabian students when learning technical courses in English due to the primary curriculum's lack of emphasis on the English language [3]. Recognizing the significance of effective language comprehension in the classroom, the article proposes an innovative approach to aid students in understanding English words and sentences – the integration of a rapid machine translation (MT) service with PowerPoint presentations (PPTs) [3]. The paper emphasizes the urgent need for a solution that seamlessly integrates with PPTs, making it a convenient tool for teachers and students alike [3]. To provide a comprehensive understanding of the proposed solution, the researchers explore the existing limitations of current MT systems, highlighting their insufficiencies in effectively meeting the specific requirements of language learning in a technical context [3]. To achieve their objective, the authors outline the design and implementation of the ETF's (Enhanced Translation Framework) client-server network architecture.The detailed explanation of the translation process using Google Cloud translation servers sheds light on the technical intricacies of the system. The client interface is introduced, empowering users to effortlessly input English words or capture images for instant translation [3]. The client then transmits the source text to the Google Cloud server, which promptly returns the translated text. Facilitating a seamless transfer of information, the server module efficiently manages the original text and the corresponding translation, storing the latter in a database for future reference and accessibility. The research paper doesn't merely stop at the technical aspects but proceeds to evaluate the performance of the ETF implementation [3]. Key elements, such as the effectiveness of the dictionary module and the overall translation process, are meticulously examined. The findings reveal promising results, showcasing that the ETF considerably reduces instruction time by aiding students in comprehending English material effectively [3]. The author envision the ETF evolving into an autonomous system, alleviating its reliance on external translation APIs, thus providing a self-sufficient and robust language translation solution for Saudi Arabian students navigating technical courses taught in English [3]. With this research paper, the path towards more inclusive and effective language education takes a significant leap forward, promising brighter prospects for students in Saudi Arabia and potentially inspiring similar endeavors worldwide.

2.3 LITERATURE REVIEW III

The Author aims to create a mobile application for Indonesian and Madurese translators using RESTful API with JSON data format.In order to build a translator system that can be used by all platforms, including Android, a web service must be created. Web service is a standard and a programming method for sharing data between several applications [4]. This paper talks about language translator where most of the population don't understand language and area unit unable to speak effectively with the deaf [4]. Therefore, the deaf realize it tough to converse with folks on daily to day basis, this issue are often solved through a smartphone application. This research work proposes a portable and 24x7 available system with support for bidirectional translation i.e. from sign language to speech and speech to sign language [4]. The mobile application will give normal speech output as audio and text and sign language output as a 3D animated video sequence, with the help of Unity3D. According to the research results, there are some recommendation on this system to fulfil the needs and requirements of the end-users [4]. In future, new improvements can be implemented on this application where the upgraded versions can provide the user to access more languages for translation.Moreover, online functions can be added to provide more updated information. This device basically can be used by people who do not know English and want it to be translated to their native language [4]. It involves extraction of text from the image and converting the text to translated speech in the user desired language [4]. In this paper, authors developed and introduced an Android- based framework that translates the American Sign Language to a text that can be used anywhere. The mobile camera shots the picture, and skin segmentation is achieved using YCbCr systems [4]. Features are extracted from the image using HOG and list to recognise the symbol. In this paper, author developed an English to Igbo Language Translation Natural Language Processing System in Android [4]. The Design Word, Reference System, and Decoder were performed in Microsoft Hub. The new English Text to Multilingual Speech Translator using Android (T2MSTA) is designed to help people who lack the power to talk or non-native speakers and individuals who do not share a common dialectal. In this paper the author talks about Android Platform for Machine Translation -A Focus on Yorùbá Language Which was developed on a mobile platform for easier accessibility,convenience, and portability.

2.4 LITERATURE REVIEW IV

The purpose of this research is to compare eight well-liked, free, online machine translation systems employing seven distinct languages (Chinese is not the target language) [5]. The translation accuracy of these systems was evaluated by the researchers using automated evaluation methods including the BLEU score [5]. According to the study's findings, Google Translate outperformed the other artificial language translators relative to its total performance. The paper's line of reasoning and supporting data are strongest when compared comprehensively across different machine translation systems in various languages [5]. The study offers a rapid and affordable method to evaluate translation accuracy that has a strong correlation with human review by employing automatic evaluation methods like the BLEU score. The report also emphasizes the significance of contemporary automatic language translators, whose translation abilities have greatly improved [5]. The study does have significant shortcomings and restrictions, though. First, the study forgoes human review of translations in favor of machine grading techniques, which might miss some subtleties or cultural background [5]. Second, there are just six languages included in the study's sample of texts and languages. The performance of these translation algorithms could be better understood by increasing the sample size and including a wider range of language pairs [5]. The study also emphasizes free internet translators, which could not accurately reflect the performance of translation tools that are sold commercially [5]. Vanjani and Aiken's study compares eight well-known, free machine translation tools employing seven different languages. The researchers offer insights into these systems' overall performance as well as their respective strengths and limitations by assessing these systems across a variety of languages and language pairs [5]. The paper concludes the article offers useful insights into the effectiveness of free online machine translation systems, particularly when compared to Google Translate [5]. The generalizability of the results may be constrained by its reliance on computerized evaluation and scant language coverage. To provide a more thorough knowledge of the capabilities and restrictions of machine translation systems, future research in this area should take these restrictions into account.

2.5 LITERATURE REVIEW V

The study "Neural Machine Translation: A Comprehensive Review" offers a thorough analysis of the literature on the improvements and developments in Neural Machine Translation (NMT). This review seeks to provide an overview of the major contributions and turning points in the field of NMT, which has become the dominant paradigm in machine translation [6]. The introduction of the Transformer model, which has evolved into the de facto architecture for NMT, is covered in the first section of the paper. The model's self-attention mechanism, which the authors highlight, allows it to effectively capture word dependencies [6]. Additionally, they go over the significance of pre-training and fine-tuning techniques, which have greatly enhanced translation quality. The paper then goes in-depth on multilingual and low-resource NMT improvements [6]. It examines strategies like transfer learning, in which models honed for languages with high resources are used to enhance translation for languages with low resources [6]. The authors emphasize the effectiveness of multilingual models that are capable of managing many languages at once, facilitating information sharing and raising translation quality between various language pairings. The research looks at several enhancements to attention mechanisms, which are important in NMT [6]. It talks about the introduction of several attention mechanisms, including self-attention and cross-attention, and how they affect the caliber of translation [6]. The authors also go through attention-boosting strategies that have improved NMT model performance and interpretability, such as adding syntactic data and linguistic structures. The review emphasizes the value of comprehensive evaluation initiatives. Through these activities, benchmarks and standardized evaluation measures for contrasting various NMT models have been made available [6]. The importance of these assessments in developing the field and enabling fair comparisons between various techniques is emphasized by the authors [6]. The paper concludes with a thorough literature analysis on NMT that covers the main developments in the area. It draws attention to the significance of the Transformer model, developments in low-resource, multilingual NMT, enhancements to attentional mechanisms, and extensive evaluation campaigns [6]. Researchers and practitioners interested in learning about the developments and trends in neural machine translation will find this review to be a useful resource.

3 METHODOLOGY

3.1 Technologies for Language Translator

Some of the key technologies commonly used in language translation projects include:
Tkinter: A common Python library for graphical user interface (GUI) development. It offers numerous widgets and development tools for GUI applications. The primary program window, labels, text widgets, buttons, frames, and other GUI elements are all created in this code using Tkinter.

ttk: In Tkinter, a themed widget set called ttk was introduced. It offers upgraded and stylish GUI components, like comboboxes. The code generates comboboxes for choosing the input and output languages using the ttk module.

CSV datasets: The datasets are made in a csv format. There are five different datasets to support five different language translations.

3.2 Theoretical formulation

The methodology for the development of the Language Translator involves several theoretical formulations and principles to guide the implementation. These formulations outline the key steps and considerations for building a flexible and functional Language Translator using Python. The following theoretical formulations can be applied:

1. Import the required modules: The tkinter, ttk, and CSV modules are imported to provide the necessary functionality for GUI development and text translation.

2. Define the translation function: The translatetext() function is defined to handle the translation process. It helps in the translation of the text in a English language to the target language by using the dataset of CSV formats.

3. Create the main application window: The Tk() function is used to create the main window of the application. The window's properties, such as size, title, and resizing capability, are set accordingly.

4. Create the GUI elements: Various GUI elements are created using Tkinter. These include labels, text widgets for input and output, comboboxes for language selection, and a translate button. The elements are packed and positioned within frames to organize

the layout.

5. Configure event handling: The translate button is assigned the translatetext() function as its command, which will be executed when the button is clicked.

6. Start the main event loop: The mainloop() method is called to start the main event loop, which listens for user interactions and updates the GUI accordingly.

3.3 Mathematical Modelling

The mathematical modelling of the language translator follows the given equation : Y = A X + B

here,

Y is the output language

X is the input language

A is the turning point

and B is the constant

The obtained output language is determined by the turning point. This involves the selection of Input and Output Language by the user. Similarly, the output language depends upon the input language and selection of the input and output language parameter.

3.4 System Block Diagram

The system block diagram for the language translator project is shown in Figure 1. The input will be the text entered by the user. Once the text is entered, the user selects the language that the entered text is written in. After selecting the source language, the user will select the target language. Once the target language is selected, the user presses the translate button. On doing so, the user can get the translated text in the output box.

Interface: The user interacts with the interface to perform the translation.

Text Input: The user provides the input text through the input column. The text should be in English.

Target Language: The user then selects the target language for the translation. We have included five different languages namely Spanish, German, French, Japanese and Nepali.

Translation: The translate button provides the translation by getting the target text from

the appropriate CSV files as per the language selected.

Translated Text: The final step is the translated text that we get from the CSV files. It is also the desired output by the user.

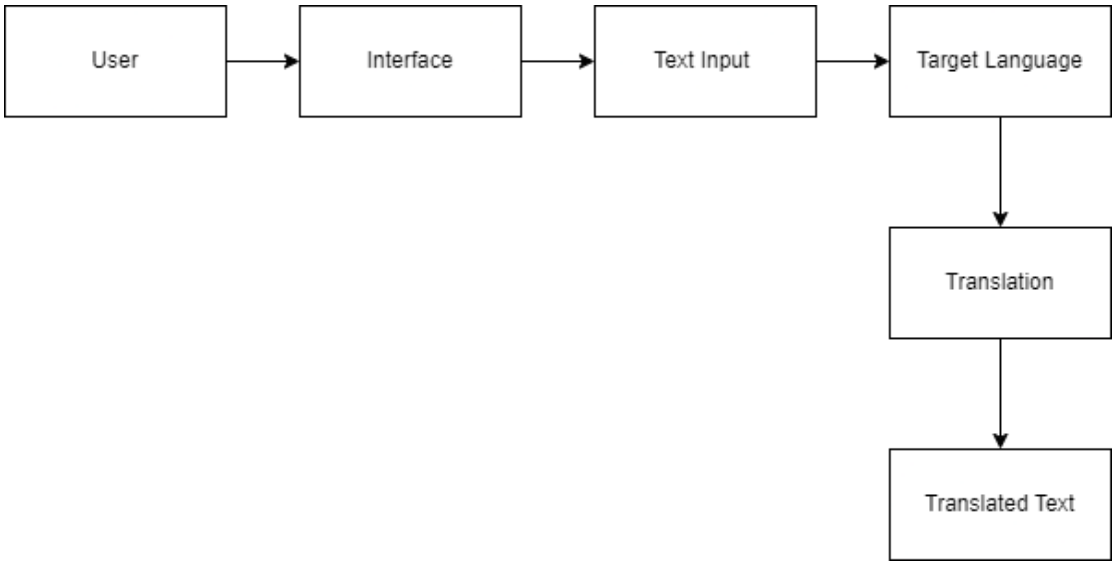

Figure 3.1: System Block Diagram

Figure 3.1 shows the intended system block diagram of the proposed system.

3.5 Instrumentation Requirements

A graphical user interface (GUI) for a language translation program is provided by the code. The user can type text in a text widget for input, choose the input and output languages from comboboxes, and read the translated content in a different text widget. The CSV dataset is used to implement the translation feature. Setting up the GUI elements, creating a translation function that makes use of the Translator class in the library, and handling user interactions to start the translation process are all parts of the algorithm. Upon user input or language selection, the translated result is shown in real-time. The Tkinter library is used in the code to manage layout and create GUIs.

Hardware	Software	Tools/IDE	Programming Language
A modern computer with decent processor and RAM	Python 3.11.4	Visual Studio	Python
A monitor or display screen	Tkinter		
	PIL (Python Imaging Libraries)		
	CSV Files		

Table 3.1: Instrumentation Requirements

3.6 Feasibilty Analysis

Feasibility Analysis for Language Translator Project are:

1. Technical Feasibility:

The technical feasibility of the Language Translator project is high. The code is built using Python and utilizes the tkinter library for creating the GUI. Both Python and tkinter are widely used and well-established technologies, making it easy for developers to understand and modify the code. Additionally, the translation process relies on simple word-to-word mapping from CSV datasets, which is a straightforward approach. However, for more advanced features, such as integrating NMT models or real-time translation APIs, developers would require expertise in machine learning and NLP technologies.

2. Economic Feasibility:

The economic feasibility of the project is relatively low due to the reliance on a limited CSV dataset for translations. While CSV datasets are cost-effective, they may not provide comprehensive translations for all words and phrases. Enhancing the dataset or integrating external translation APIs with a cost associated could improve translation accuracy but may require financial investment. Moreover, if the project is intended for commercial use, additional costs may arise for licensing language models, hosting servers, and maintaining the application.

3. Legal Feasibility:

The legal feasibility of the project is generally favorable, provided that the CSV datasets used for translations are obtained and used legally. Care should be taken to ensure that the datasets do not violate copyright or intellectual property rights. If the project aims to collect user-provided translations for dataset enhancement, proper consent and data protection measures must be implemented to comply with privacy laws and regulations.

4. Operational Feasibility:

The operational feasibility of the project is reasonably high. The GUI interface and simple translation process make the application easy to use for non-technical users. However, the limited dataset may result in occasional errors or missing translations, affecting user experience. To enhance operational feasibility, continuous maintenance and updates to the dataset can be implemented to improve translation accuracy and

expand language support.

5. Schedule Feasibility:

The project's schedule feasibility depends on the scope and complexity of planned enhancements. Integrating more advanced language models, expanding datasets, and implementing user feedback mechanisms would require substantial development time. The project's feasibility in meeting deadlines and milestones relies on the availability of skilled developers and resources.

3.7 Dataset Explanation

We have used CSV files to prepare five different datasets in our project for the five different languages translation. These files are imported in the code and used based on the languages selected in the combobox by the user. The datasets are fully developed by us and has limited words for now. We can add as many words as we want in the datasets. By doing so, we can make a rich datasets in the near future.

Below is the explanation of the datasets used in the provided code:

1. **en to es.csv (English to Spanish Dataset):** This CSV file contains word pairs for translation from English to Spanish. Each row in the file represents a word or phrase in English and its corresponding translation in Spanish.

2. **en to fr.csv (English to French Dataset):** This CSV file contains word pairs for translation from English to French. Each row in the file represents a word or phrase in English and its corresponding translation in French.

3. **en to de.csv (English to German Dataset):** This CSV file contains word pairs for translation from English to German. Each row in the file represents a word or phrase in English and its corresponding translation in German.

4. **en to ja.csv (English to Japanese Dataset):** This CSV file contains word pairs for translation from English to Japanese. Each row in the file represents a word or phrase in English and its corresponding translation in Japanese.

5. **en to ne.csv (English to Nepali Dataset):** This CSV file contains word pairs for

translation from English to Nepali. Each row in the file represents a word or phrase in English and its corresponding translation in Nepali.

Each of these CSV files contains a list of word pairs representing common English words or phrases and their translations in the respective languages. These datasets are used by the translator application to perform basic word-to-word translations for the supported languages. However, it is essential to note that the datasets are limited, which may result in incomplete or inaccurate translations for complex sentences or less common words or phrases. To achieve more accurate and context-aware translations, state-of-the-art NMT models are generally employed, which can handle a broader range of language nuances and sentence structures.

English Sentence.	German Sentence
Hello.	Hallo
Good Morning.	Guten Morgen
Good Afternoon.	Guten Tag
Good Evening.	Guten Abend
Nice to meet you.	SchÃ¶n Sie kennenzulernen
Thank you.	Danke
You're welcome.	Bitte
Excuse me.	Entschuldigung
I am sorry.	Es tut mir leid
Yes.	Ja
No.	Nein
Can you help me?	KÃ¶nnen Sie mir helfen
I don't understand.	Ich verstehe nicht
Could you repeat that?	KÃ¶nnten Sie das wiederholen
Do you speak English?	Sprechen Sie Englisch
I speak a little English.	Ich spreche ein wenig Englisch
I am hungry.	Ich habe Hunger
Where is the bathroom?	Wo ist die Toilette

Figure 3.2: Dataset

3.8 Description of Algorithms

The algorithm for the Language Translator is as follow:

Step 1: Import the necessary modules and libraries: tkinter for creating the graphical user interface (GUI) and ttk for additional GUI elements

Step 2: Define the translation function translate text():

a. Create an instance of the Translator class from googletrans and specify the serviceURLs.

b. Retrieve the input text from the Input text text widget.

c. Get the selected input language and target output language from the respective comboboxes.

d. Call the translate() method of the translator instance, passing the input text, input language, and target language as arguments.

e. Retrieve the translated text from the translation result.

f. Clear the previous output text from the Output text widget.

g. Insert the translated text into the Output text widget.

Step 3: Create the main application window using Tk() and set its properties: Set the window size to 800x500 pixels. Disable window resizing. Set the window title as ”Language Translator”.

Step 4: Create the GUI elements: a. Create a label for the header with the text ”Language Translator” and set its font.

b. Create a frame for input and output elements.

c. Create a frame for input elements:

i.Create a label for the ”Enter Text” prompt.

ii.Create a text widget for user input (Input text).

iii.Create a frame for the input language selection:

iv.Create a label for the ”Input Language” prompt.

v.Create a combobox for selecting the input language (input lang combo).

d. Create a frame for output elements:

i.Create a label for the ”Output” prompt.

ii.Create a text widget for displaying the translated output (Output text).

iii.Create a frame for the output language selection:

iv.Create a label for the ”Select Language” prompt.

v.Create a combobox for selecting the output language (dest lang combo).

e. Create a ”Translate” button with the command set to the translate text() function.

Step 5: Pack and position the GUI elements in the window using appropriate layouts and positioning methods.

Step 6: Start the main event loop using mainloop(), which listens for user interactions and updates the GUI accordingly.

3.9 Elaboration of Working Principle

The Language Translator project exhibits a high level of technical feasibility as it utilizes well-established technologies like Python and tkinter for the user interface. However, to enhance the translation capabilities further with advanced language models, developers would require expertise in machine learning and natural language processing. In terms of economic feasibility, the current reliance on a limited CSV dataset might present cost-effectiveness, but incorporating more comprehensive datasets or external translation APIs could potentially improve accuracy, albeit with associated costs. From a legal perspective, the project appears to be on solid ground, provided that datasets are obtained and used legally and user data is handled with proper consent and privacy measures. Operational feasibility is promising due to the user-friendly GUI, making it accessible to non-technical users, yet ensuring optimal user experience would necessitate continuous maintenance and updates, especially with dataset improvements. The project's schedule feasibility would rely on available resources and the scope of development and enhancements. In conclusion, the Language Translator project is technically viable with its current implementation, but to maximize its potential, ongoing improvements and advancements in language translation technologies are essential to ensure accurate and versatile translations.

3.10 Verification and Validation Procedures

User Evaluation of Language Translator: Users evaluate language translator as a means for translating languages. The Language translator can be evaluated as an easy interface to translate languages from English to five different languages. The translation is not a rocket science and can easily be done with the help of just a few clicks on the interface. The user have evaluated the translator to be easy and fun experience to learn various languages and have given a good response to the project. The system is verified by the users and supervisor.

System Evaluation of Language Translator: The system evaluation of the Language Translator focuses on the technical part of the translator. The Language Translator connects to the data-set to the Python code and help provide the translations given in the different CSV files.The effectiveness of the system depends upon the data-set. For now, the data-set is limited so it might not provide all the required translation, but we will enhance it in the future and make a rich data-set. The project is valid and meets the required planning done in the initial phase.

3.11 SDLC Method for Project

Our Language Translator project follows the Rapid Application Development (RAD) methodology, which is well-suited for short projects like ours. RAD emphasizes rapid prototyping, iterative development, and close collaboration with stakeholders to deliver a functional system quickly.

Requirements Gathering:

During the initial stage, our team conducted research and discussions to select the topic for the project, which led us to the Language Translator idea. We gathered requirements for the project, including the selection of Python libraries for GUI (tkinter) and image processing (PIL), as well as the identification of datasets required for translations.

Planning:

In the planning phase, we laid out a clear scope for the project. We defined the schedule, budget, and allocated resources for development. We decided to use the tkinter library for building the GUI-based interface. Additionally, we discussed the mechanism for translation using CSV datasets. We also identified potential risks and ways to mitigate them.

Rapid Prototyping:

The core concept of RAD is rapid prototyping, and we embraced this approach. We developed an initial prototype of the Language Translator, which allowed us to gather valuable feedback from both users and our supervisor. Based on the feedback received during the mid-defense and first defense, we refined and modified the prototype to improve its functionality and usability.

Iterative Development:

Following RAD's iterative development process, we moved on to build the Language Translator incrementally. Each iteration built upon the previous one, incorporating feedback and enhancements. This iterative approach allowed us to make continuous improvements to the translator and cater to specific user needs.

Testing:

A critical phase in the RAD methodology is testing. We thoroughly tested the Language Translator to ensure it met the specified requirements. Functional testing was carried out to verify the accuracy of translations. We also conducted performance testing to assess the application's speed and responsiveness. Additionally, security testing was performed to identify and address potential vulnerabilities.

Deployment:

With our robust testing process complete, we proceeded to deploy the Language Translator. We carefully checked the entire system to ensure smooth functionality in the operational environment.

Maintenance:

The RAD approach acknowledges that software development is an ongoing process. Therefore, we are committed to providing continuous maintenance and support for the Language Translator even after deployment. This includes addressing any potential bugs or issues that may arise and making improvements based on user feedback.

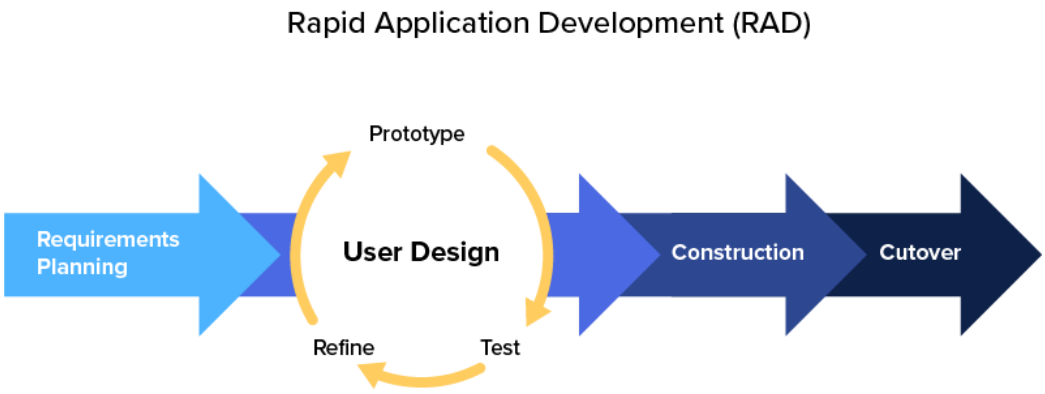

Figure 3.3: RAD Model

3.12 Project Design and Procedures

DFD

DFD of the Language Translator shows the flow of data from one place to the other in the project. It helps us understand the project in more detail and understanding the working or atleat get a simple understanding of the system. The Level 0 and Level 1 DFD of the Language Translator is given below. Here the data flows between the user and Admin in the DFD 0 while the databse are also included in the DFD level 1. Also the processes are further divided into source language, target language and translation in the DFD 1 as compared to the DFD 0 where only language translation is used as process.

DFD Level 0

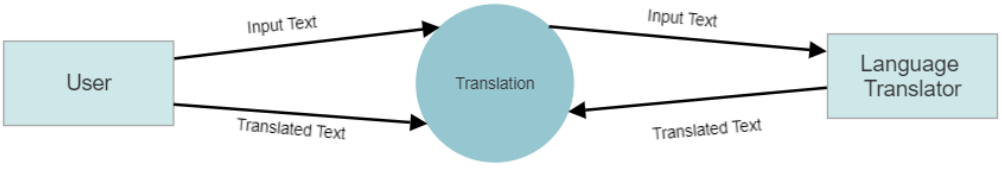

Figure 3.4: DFD Level 0

DFD Level 1

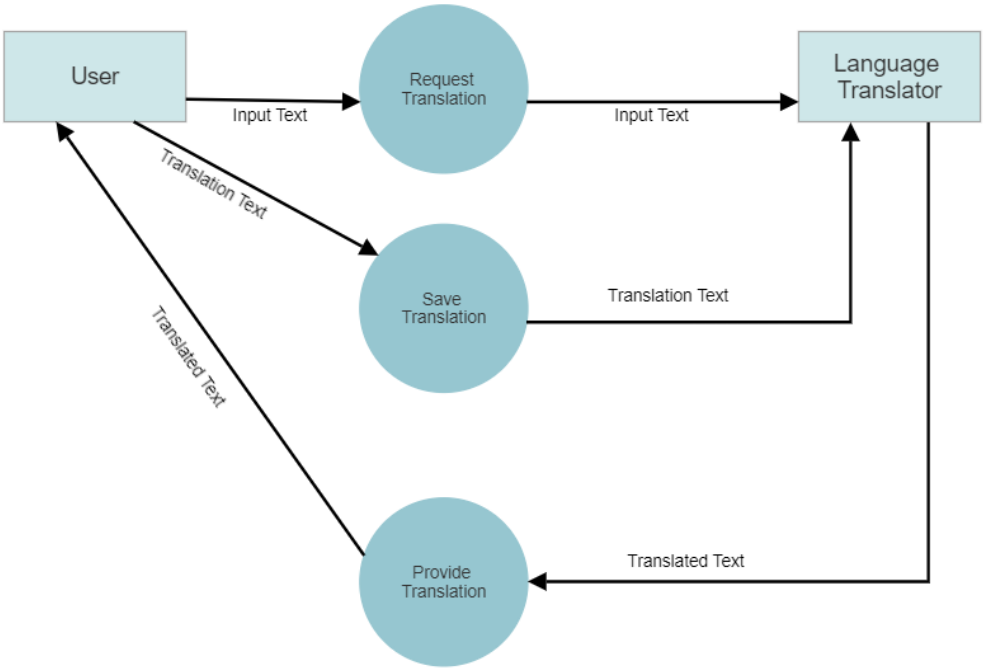

Figure 3.5: DFD Level 1

DFD Level 2

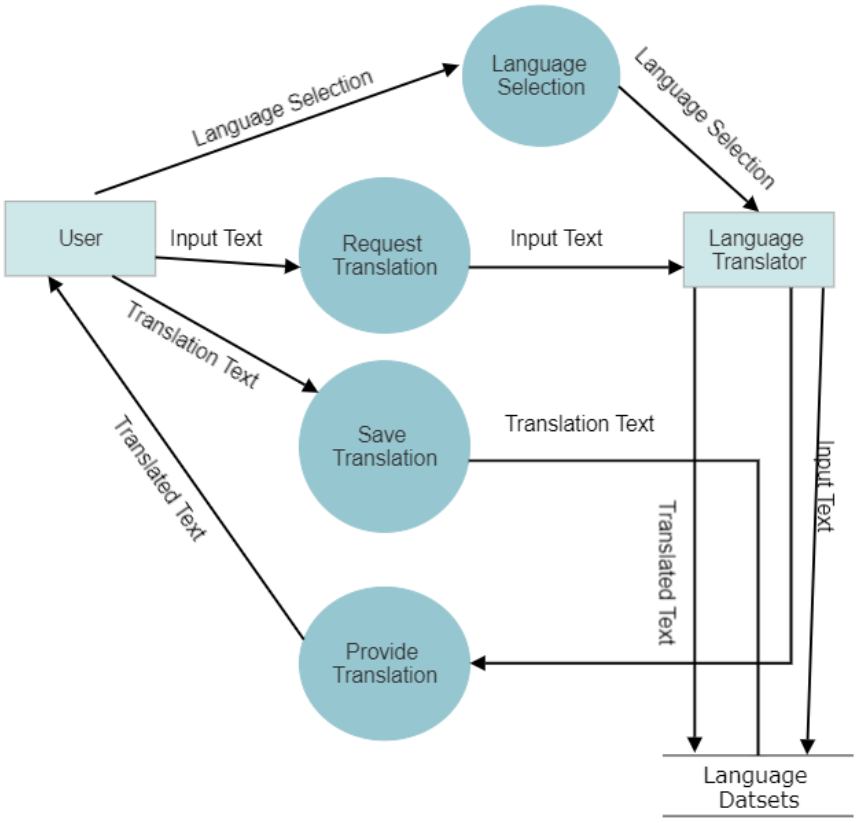

Figure 3.6: DFD Level 2

Use Case

Use case diagram of the Language Translator has two actors named User and Translator. The user has access to request translation, language options, language selection and get translation. Similarly, the translator has the access to determine the source language, confirm or reject, provide translation and access database

UseCase

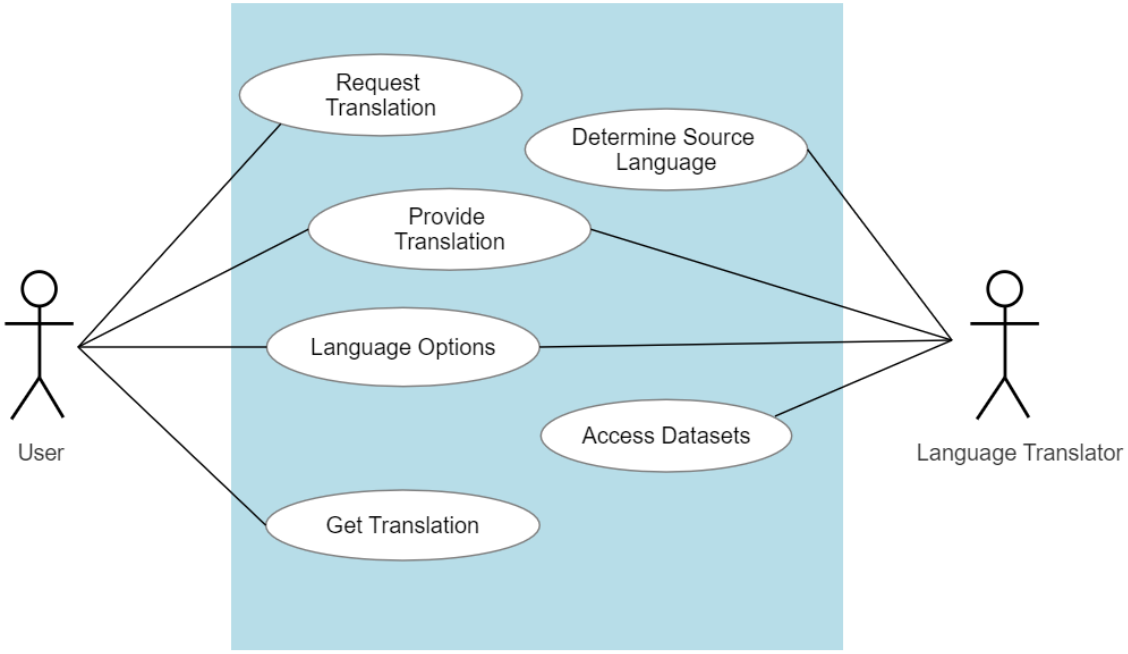

Figure 3.7: Use Case

4 RESULT

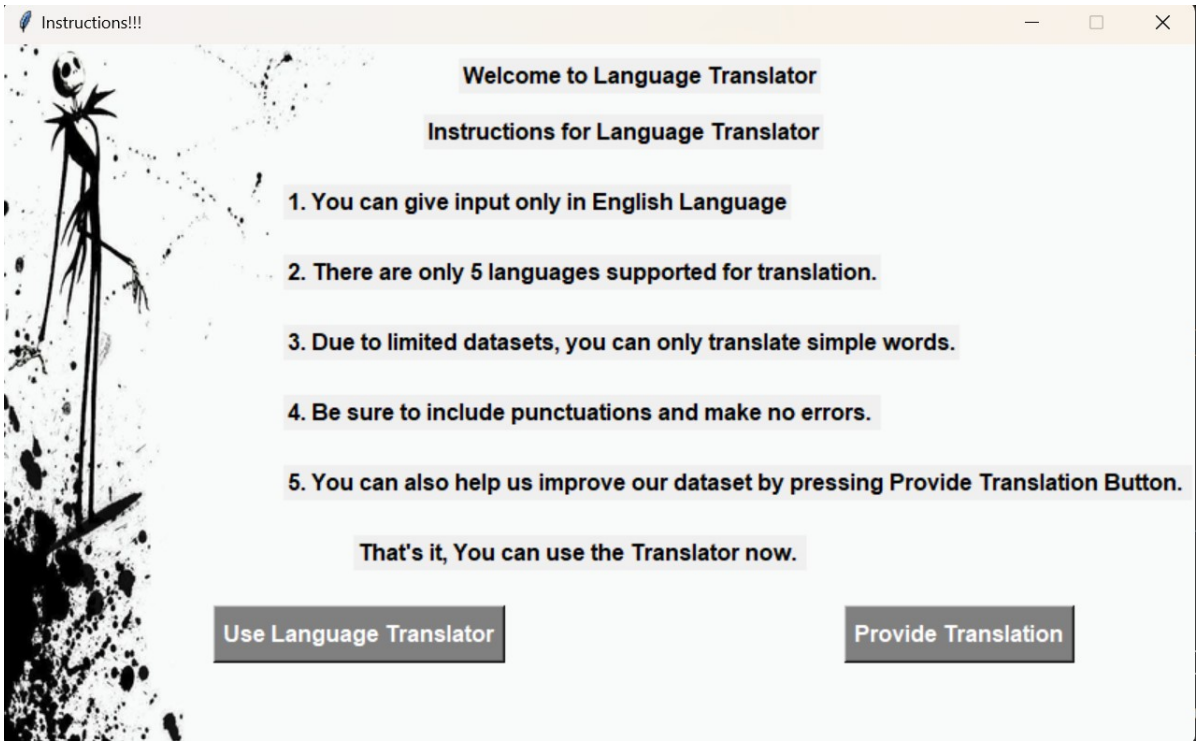

Figure 4.1: Result

This is the first interface that you get when using our Language Translator. Here, we have provided some guidelines or instructions that the user must follow in order to have the most efficient use of the Translator. Then, we have two buttons Language Translation and Provide Translation. When clicking on the Language Translation button the user gets another interface as shown in the figure 4.2

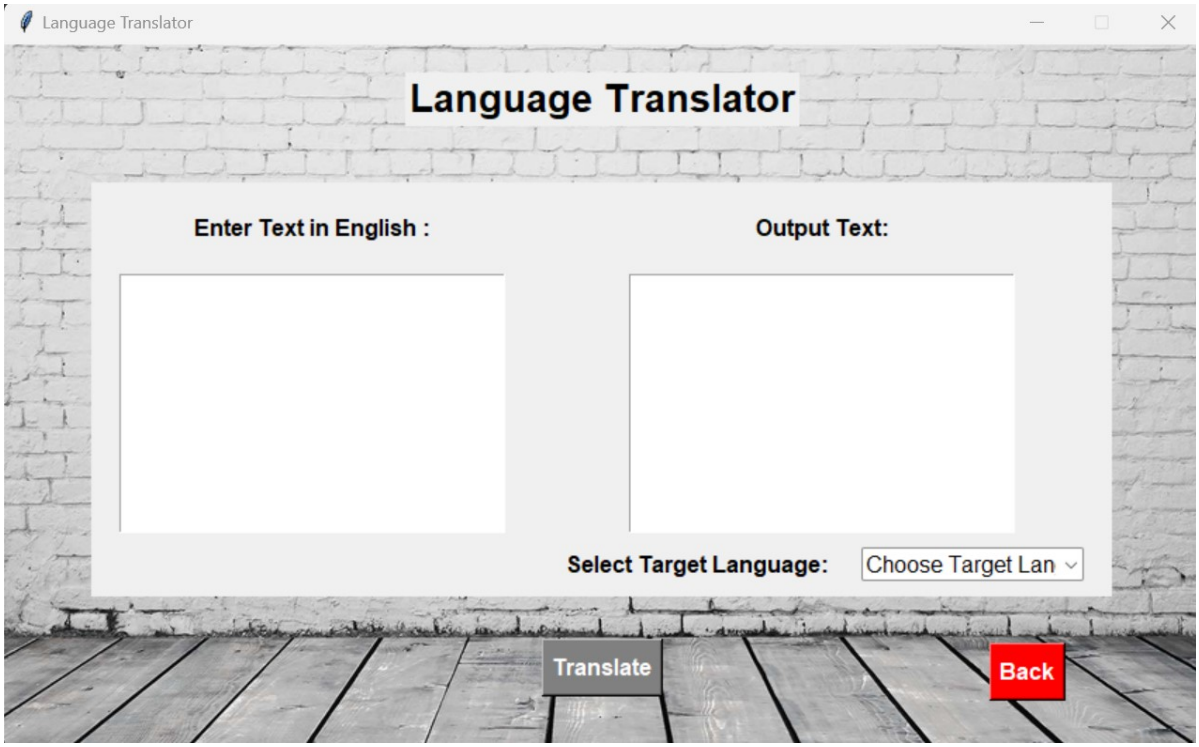

Figure 4.2: Result

This is the interface that allows you to get the translations of the text you want. The user can enter the English text in the Input Section and choose the target language before clicking the translation button. After clicking on the Translate Button, the user gets the required translation. If the user happens to enter the text which has not been added to the data set, the user gets an error message. After the translation, we can click the back button to go to previous page or just close the interface.

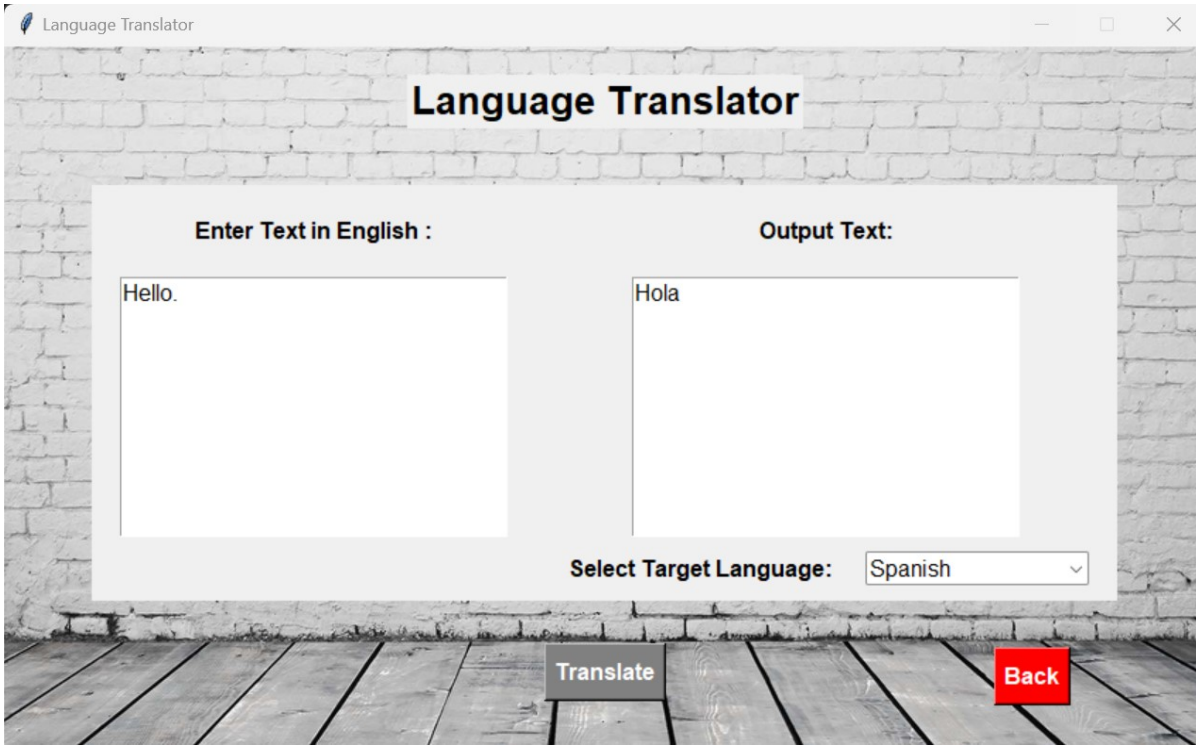

Figure 4.3: Result

Now, this is the result that the user gets, when he translates a simple text ”Hello.” into Spanish. The translated text is shown in the output section as ”Hola” which is the spanish for Hello. This also shows the accuracy of our translator.

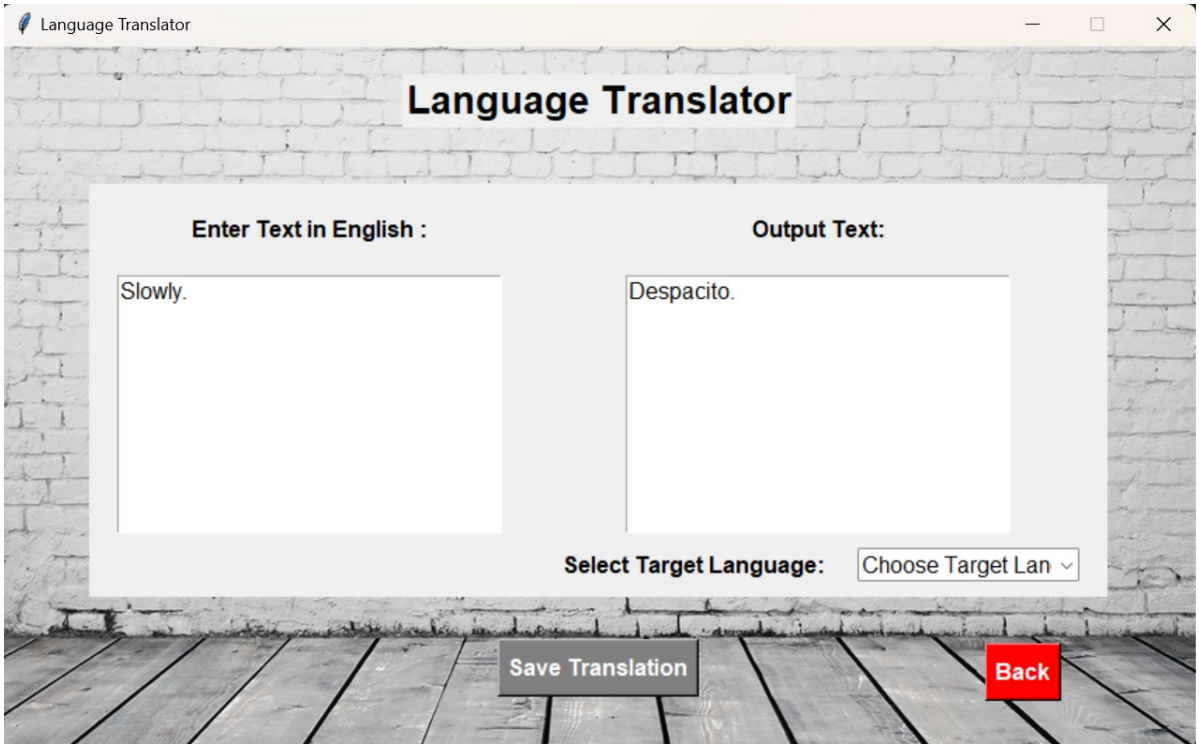

Figure 4.4: Result

Say, the user knows a certain term in any of the five languages but we have not yet included the text in our dataset, then the user can help us by adding the translation in our data set. To do that the user just need to click on the Provide Translation Button, On doing so, the user gets a similar interface as the Language Translation only this time there is an interface to provide both the input and output text. After providing the Input and Output text , the user can simply add the translation in the dataset by selecting the target language and clicking the Save Transaltion button.

Here, the user has added a text slowly in English that translates to Despacito in Spanish. After selecting the Spanish language and clicking the Save Translation button, the user saves the translation in the dataset.

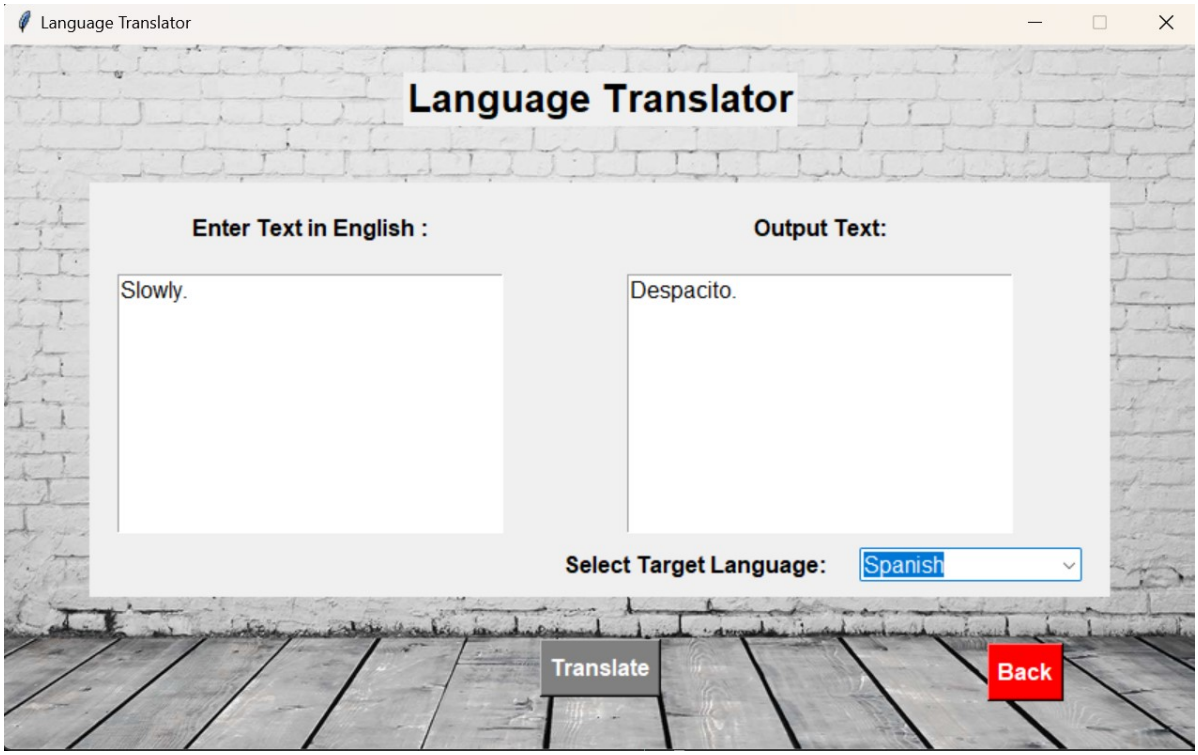

Figure 4.5: Result

Here, we didn't have the word Slowly in our dataset before, but after the user provided the translation, we can translate the word whenever a new user enters the word. The result is shown in the figure 4.5

5 DISCUSSION AND ANALYSIS

The project uses a basic GUI-based interface implemented using the tkinter library in Python. The translator allows users to input text in English and translates it into five supported languages: Spanish, French, German, Japanese, and Nepali. The translator utilizes CSV files containing word pairs for each language to perform the translations. Additionally, the translator provides an option for users to contribute translations to improve the dataset.

1. Theoretical vs. Simulated Outputs: The theoretical output refers to the expected translation based on the provided dataset (CSV files), while the simulated output is the actual translation produced by the translator using the same dataset. The theoretical outputs are based on the assumption that the provided dataset contains accurate and comprehensive translations for all input words.

Discrepancies between theoretical and simulated outputs may arise due to several reasons:

a) Limited Dataset: The provided dataset might be limited, leading to missing translations for certain words or phrases, resulting in ”No translation found in the dataset” error messages.

b) Ambiguity: Some English words or phrases may have multiple translations in the target language, and the dataset might not cover all possible translations.

c) Dataset Quality: The accuracy and completeness of the dataset can significantly impact translation quality. If the dataset contains errors or incomplete information, the translations might not be accurate.

d) Handling Complex Text: The code seems to be designed to handle simple words or phrases, and complex sentences might not be translated accurately due to the absence of proper grammar rules and context understanding.

2. Error Analysis: Possible sources of error in the project include:

a) Incorrect Dataset: If the CSV files used for translations contain incorrect or insufficient data, it can lead to inaccurate translations or error messages.

b) UI Issues: The GUI might not handle all possible user input scenarios, such as empty

input or invalid characters, leading to unexpected behavior.
c) Hardcoded Paths: The code uses hardcoded file paths for dataset files, which can cause issues if the files are moved or the paths are not correct.

3. Tally with State-of-the-Art: As the translator does not include a state-of-the-art translation model like modern Neural Machine Translation (NMT) systems, it is not directly comparable to the latest language translation methods. The translation process in this code relies on basic lookup from a CSV dataset, which is limited in scope and accuracy.

4. Methodology Performance: The methodology used in this code is quite simple and may perform adequately for basic word-to-word translations within the supported languages. However, it has several limitations:
a) Limited Scope: The translation capabilities are restricted to a small dataset and may not cover all possible translations for a given word or phrase.
b) Accuracy: The accuracy of translations heavily relies on the quality and comprehensiveness of the dataset. In the absence of a large, diverse dataset, the translations may not be reliable.
c) Grammar and Context: The code lacks advanced language models that can handle grammar, syntax, and contextual information, which are essential for accurate translations of complex sentences.

While the translator offers a basic implementation of a Language Translator using a CSV dataset, it has inherent limitations in terms of accuracy and scope. To achieve state-of-the-art translation performance, advanced NMT models like Transformer-based architectures with extensive training on large bilingual datasets are generally employed. These models can learn complex patterns, sentence structures, and contextual information, leading to more accurate and contextually-aware translations. To improve the methodology, one could consider integrating a pre-trained NMT model, using online translation APIs, or accessing more extensive and diverse datasets for better translation results.

6 FUTURE ENHANCEMENTS

Future Enhancements for the Language Translator Project project consists of various factors to be considered, some of which are:

1. Project Objectives Yet to be Fulfilled:

a) Expanded Language Support: Currently, the translator supports only five languages. A future enhancement could involve expanding the language support to cover more languages, making it more versatile and useful for a broader audience.

b) Improved Translation Accuracy: Implementing advanced language models like Neural Machine Translation (NMT) or transformer-based architectures can significantly enhance translation accuracy and handle complex sentence structures with context awareness.

c) Real-time Translation: Integrating real-time translation capabilities could allow users to input continuous streams of speech or text, making the application more dynamic and user-friendly.

d) User Profiles and History: Introducing user profiles and history tracking can enable personalized translations and maintain a record of past translation requests for reference.

2. Course of Action to Complete Remaining Tasks:

a) Implement Advanced NMT Models: Research and incorporate state-of-the-art NMT models, such as Transformer-based models, using popular libraries like TensorFlow or PyTorch. These models can be pre-trained on large bilingual datasets for improved translation accuracy and context understanding.

b) API Integration: Integrate popular translation APIs, such as Google Cloud Translation API or Microsoft Translator API, to leverage their robust language models and support for numerous languages. This would allow the application to benefit from ongoing updates and advancements in translation technologies.

c) User Feedback Mechanism: Add a user feedback mechanism to collect feedback on translations, enabling continuous improvement of the translation dataset and model performance.

d) Context Awareness: Enhance the translation logic to consider context and previous user inputs, ensuring more accurate and contextually relevant translations.

e) UI Enhancements: Improve the user interface with a more intuitive design, real-time

feedback during translation, and error handling for better user experience.

f) Speech-to-Text Input: Implement speech-to-text conversion, allowing users to provide input through voice, thereby making the application more accessible and user-friendly.

g) Cloud Storage: Utilize cloud storage for datasets and user profiles, ensuring scalability and accessibility from multiple devices.

3. Testing and Validation:

Thoroughly test the enhanced application with various input scenarios and datasets. Compare the translations against known references and evaluate the accuracy and performance of the integrated NMT models and translation APIs. Gather user feedback through beta testing and address any issues or suggestions for further improvement.

4. Documentation and User Guides:

Create comprehensive documentation and user guides explaining the enhanced features, system requirements, and usage instructions. Provide examples of supported languages and translation outputs to assist users in maximizing the application's potential.

By pursuing these future enhancements, the Language Translator project can evolve into a more sophisticated, accurate, and user-friendly tool, catering to a broader range of users and language translation needs.

7 CONCLUSION

The Language Translator project represents a commendable effort in creating a basic GUI-based tool for language translation using Python and the tkinter library. The application enables users to input English text and obtain translations in five supported languages, which can be beneficial for cross-cultural communication and understanding. However, upon analysis, it becomes evident that the project has certain limitations that need to be addressed to enhance its overall performance and accuracy. The theoretical vs. simulated outputs analysis revealed potential discrepancies arising from a limited dataset and the absence of advanced language models. These discrepancies may lead to inaccurate translations, particularly for complex sentences and technical terminology. Additionally, error analysis identified possible issues related to dataset quality, user interface handling, and hardcoded file paths, which could impact the overall user experience and reliability of the translations. While the project offers a basic translation service, it falls short when compared to state-of-the-art language translation systems that employ advanced Neural Machine Translation (NMT) models with extensive training on large bilingual datasets. The project's methodology, relying on a CSV dataset, constrains the scope of translations and may not cover all possible language variations. We had four objectives in the beginning and we were able to fulfill 3 out of them so the objective fulfilled can be given as:

$$\frac{(4-1)x100\%}{4} = 75\%$$

Despite these limitations, the Language Translator project serves as a valuable starting point for language translation development. To advance the project's capabilities, future enhancements are crucial. Integrating pre-trained NMT models, incorporating online translation APIs, and accessing more comprehensive and diverse datasets would significantly improve translation accuracy and broaden language support. The project demonstrates technical feasibility and provides a foundation for further improvements. To make the application more robust and competitive with modern translation tools, continuous development, updates, and advancements in language processing technologies are vital. With dedication to refining the methodology and expanding the dataset, the Language Translator has the potential to become a reliable and versatile tool for facilitating effective communication across languages and cultures.

APPENDIX A

A.1 Project Schedule

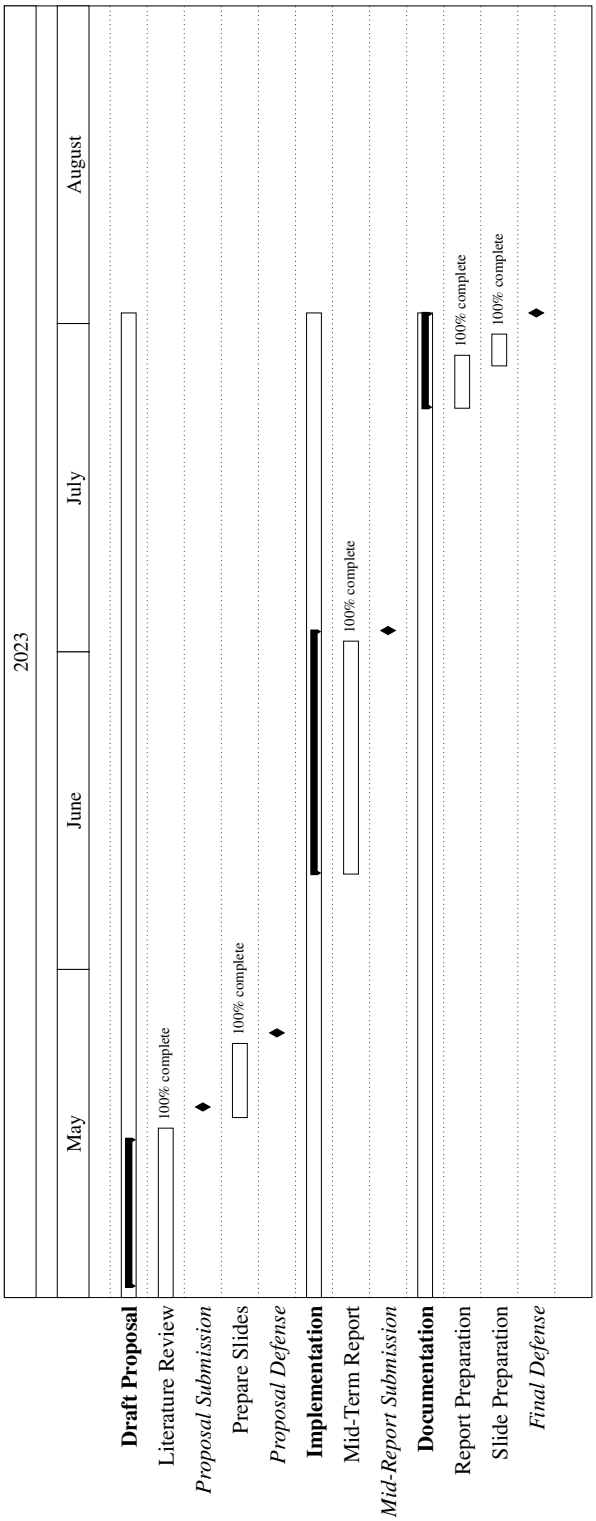

Figure A.1: Gantt Chart showing Expected Project Timeline.

,

A.2 Literature Review of Base Paper- I

Author(s)/Source: Gelavizh Abbasi, Saman Saleh zadeh, Elenaz Janfaza, Arezoo Assemi, Siamak Saadat Dehghan	
Title: Language, Translation, and Culture	
Website: t:https://www.researchgate.net/publication/349867394	
Publication Date: March 2012	**Access Date: March 2012**
Journal: 2012 International Conference on Language, Medias and Culture	**Place:** Islamic Azad University-Urmia Branch/Iran
Volume: 05	**Article Number:** 2012
Author's position/theoretical position: student , student , student, student, student,	
Keywords: language, culture, translation bridge, censorship	

Important points, notes, quotations **Page No.**

1. The variety of languages with different cultures and necessity of communications in human life caused translation to be a very effective factor in communicating, exchanging cultures, and knowledge. **1**
2. Although there might be is no specific frame and force on how a translation must be done, all translators who transfer natural meaning based on the cultural and religious norms of their society, encounter some limitations and censorship through translation. **2**
3. As languages and culture are complimentary of each other, and cultural features of every region is different, translators not only should concentrate on how to convey the same meaning, but also attempt to show the dissimilarities between two cultural perspectives. **2**

Essential Background Information: This paper aimed to represent some barriers in the process of translation. A good translator should simultaneously be aware of the cultural factors, views and tradition in order to consciously consider the chronological orders, explicit meaning, development of related disciplines, historical and religious background of the source text.

Overall argument or hypothesis: Language may refer either to the specifically human capacity for acquiring and using complex system of communication or to a specific instance of such a system of complex communication.

Conclusion: In conclusion, it can be pointed out that the trans-coding process should be focused not merely on language transfer but also - and most importantly - on cultural transposition. As an inevitable consequence of the previous statement, translators must be both bilingual and bicultural if not multicultural. In this study we discussed on the translation obstacles through the transferring culture among languages.

Supporting Reasons

1. A proper translation makes the reader to ponder over the cultural context of the source language.

2. Translation seems to be the only possible way to "unite" all cultures in order to create cultural network, cultural globalization

3. It acts as a bridge to communicate all kinds of languages specially those similar to each other considering their linguistic features and cultural customs in all parts of the world.

4. The paper also define that translation has an important role in globalization of cultures especially pop culture leading to expanding ethnic habits and customs, or religious values

Strengths of the line of reasoning and supporting evidence: Difficulties arising out of differences of cultures constitute the most serious problems for translators and have produced the most far-reaching misunderstanding among readers.Culture itself has its own limitation in transferring the source text into target text.

Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: Limitation and censor as obstacles that limit translators in conveying the "semantic" message in the receptor language. [9] The other limiting factor which translators encounter is moral filtering being based on religion, family, society rules, and 84 culture, etc.

,

A.3 Literature Review of Base Paper- II

Author(s)/Source: Manjur Kolhar, Abdalla Alameen	
Title: Artificial Intelligence Based Language Translation Platform	
Website: https://www.researchgate.net/publication/350555260_Artificial_Intelligence_Based_Language_Translation_Platform	
Publication Date: 1 November 2020	**Access Date: 23 January 2021**
Journal: Intelligent Automation and Soft Computing	**Place:** Prince Sattam Bin Abdulaziz University, Wadi Ad Dawaser, Saudi Arabia
Volume: 28	**Article Number:** 1-9
Author's position/theoretical position: student, student, Prince Sattam Bin Abdulaziz University	
Keywords: Network, information science, machine learning, multimedia, server, client, language translation, application programming	
Important points, notes, quotations **Page No.** 1. The proposed framework was constructed with the aid of AutoML, which is a Google translation module **1** 2. The ETF system affords four advantages.First, auto-detection language is supported: when a user does not know the source language, the ETF automatically identi[U+FB01]es it. **1** 3. It is error-free and flexible in terms of choosing the proper translations of words. **2**	
Essential Background Information: Active learning concepts should be used for teaching when the native languages of teachers differs from students.	
Overall argument or hypothesis: The platform was mostly used for Arabic-English Translation	
Conclusion: Currently, the proposed system uses a pre-trained model from Google Translation API; hence, it cansupport many languages, including English and Arabic. The proposed framework was constructed withthe aid of AutoML, which is a Google translation module. The ETF system affords four advantages.First, auto-detection language is supported: when a user does not know the source language, the ETFautomatically identi[U+FB01]es it. Second, the ETF has glossary support: the glossary of Google Translation APIallows the storage of translated words and sentences.	
Supporting Reasons **1.** Auto detection language is supported, when a user doesn't know the source language, the ETF automatically identifies it. **3.** Batch-wise translation is possible: This allows the network administrator to reduce the network load of translating large input files.	**2.** The ETF has glossary support: the glossary of Google Translation API allows the storage of translated words and sentences. **4.** The use of the ETF as a centralized translation service is proposed to improve student-oriented learning outcomes.
Strengths of the line of reasoning and supporting evidence: The evaluation was conducted in a university laboratory where language translation experiments, especially on Arabic text, were performed.	
Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: In particular, most students, after completing the course, agreed that the ETF signi[U+FB01]cantly improved their understanding of the subject. Some students expressed the necessity of similar sessions in other subjects that are taught by non-native Arabic speakers.	

,

A.4 Literature Review of Base Paper- III

Author(s)/Source: M Vaishnavi, HR Dhanush Datta, Varsha Vemuri, L Jahnavi

Title: Language Translator Application

Website: //www.ijraset.com/best-journal/language-translator-application

Publication Date: 7 july, 2022	**Access Date: May, 2023**
Journal: International Journal for Research in Applied Science and and Engineering Technology	**Place:** Ballari Institute Of Technology and Management, Ballari, Karnataka, India.
Volume: 10	**Article Number:** 2321-9653

Author's position/theoretical position: B.E Student, Dept of CSE, Ballari Institute Of Technology and Management, Ballari, Karnataka, India.

Keywords: Android application, ML Kit,Firebase, Java, XML

Important points, notes, quotations **Page No.**

1. language has been a significant barrier for centuries now, and human beings have always tried to provide a solution to the issues of language translation. **1312**
2. The advantage of this application is it doesn't require internet connectivity. **1317**
3. This device basically can be used by people who do not know English and want it to be translated to their native language. **1313**
4. This traditional approach used for solving the problem of language differences has not been productive and favorable. **1312**

Essential Background Information: The new English Text to Multilingual Speech Translator using Android (T2MSTA) is designed to help people who lack the power to talk or non-native speakers and individuals who do not share a common dialectal.

Overall argument or hypothesis: System architecture shows the overall flow of the project and how the one system component is connected to other component and also the role of each component in the project.

Conclusion: The system that is being proposed, where translation will be implemented, will support all of the features including text scanning and text translation, as well as the languages that are widely spoken both in our country and around the world.

Supporting Reasons

1. In order to build a translator system that can be used by all platforms, including Android, a web service must be created.

2. The mobile application will give normal speech output as audio and text and sign language output asa 3D animated video sequence, with the help of Unity3D.

3. According to the research results, there are some recommendation on this system to fulfil the needs and requirements of the end-users.

4. Authors developed and introduced an Androidbased framework that translates the American Sign Language to a text that can be used anywhere.

Strengths of the line of reasoning and supporting evidence: Real time world contains different significant messages, labels and useful information but most of them are written in different official languages which depend on the host country. Besides that, it is inconvenient for a traveler to carry on their tasks in a foreign country if they don't understand the language used in that country.

Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: The project provides an Android application for language translation that does not use network and translates to the user specified language.

,

A.5 Literature Review of Base Paper- IV

Author(s)/Source: Mahesh Vanjani, Milam Aiken,	
Title: A Comparison of Free Online Machine Language Translators	
Website: http://ibii-us.org/Journals/JMSBI/V5N1/Publish/N5V1_4.pdf	
Publication Date: July 2020	**Access Date: July 2020**
Journal: Journal of Management Science and Business Intelligence)	**Place:** Jesse H. Jones School of Business, Texas Southern University, 3100 Cleburne Street, Houston
Volume: 05	**Article Number:** ISBN 2472-9264 (Online)
Author's position/theoretical position: student, student,	
Keywords: Automated Language Translator, Multilingual, Communication, Machine Translation	
Important points, notes, quotations **Page No.**	
1. Text from the source language is translated to text in the target language. **1** 2. Automatic Language Translators also referred to as machine translation software automate the process of language translation without the intervention of humans. **2** 3. However, modern automatic language translators have come a long way with vastly improved translation capability. **2**	
Essential Background Information: Because humans fluent in many languages are often not available to evaluate translations from systems, the studies above used only a few languages and evaluators, resulting in poor statistical reliability. To address this problem, automatic evaluation techniques such as BLEU (Bilingual Evaluation Understudy) are sometimes used.	
Overall argument or hypothesis: The BLEU score was proposed by Kishore Papineni, et al. in their 2002 paper "BLEU: a Method for Automatic Evaluation of Machine Translation".	
Conclusion: This study is perhaps the first to compare eight popular, free, online translation systems with seven languages in all combinations (except Chinese as the target). Results showed that, similar to other studies, Google Translate was more accurate overall as compared to the other seven automatic language translators used for this study.	
Supporting Reasons	
1. It is quick and inexpensive to calculate.	**2.** It is easy to understand.
3. It correlates highly with human evaluation.	**4.** It is language independent.
Strengths of the line of reasoning and supporting evidence: Automatic Language Translators generally organize content around thematic subjects to make it feasible to access or learn commonly used expressions or phrases that obey proper rules of syntax and grammar (Harris, 2012). . For example, automatic language translators catering to international travelers would organize content based on airports, hotels, restaurants and key phrases in the local language.	
Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: As with similar studies this study suffers from some limitations including the use of automatic evaluation rather than human review of translations, a limited sample of text, and a limitation of only six languages. Future research can potentially include other, and perhaps more complex, text samples, different languages and language pairs, and, alternate evaluation and analysis techniques.	

,

A.6 Literature Review of Base Paper- V

Author(s)/Source: Ms. Neeta Verma, Abhay Jain, Animesh Basak, Kshitij Bharti Saksena	
Title: Survey and Analysis on Language Translator Using Neural Machine Translation	
Website: https://www.irjet.net/archives/V5/i4/IRJET-V5I4833.pdf	
Publication Date: April 2018	**Access Date: April 2018**
Journal: International Research Journal of Engineering and Technology(IRJET)	**Place:** Inderprastha Engineering College, Uttar Pradesh, India
Volume: 05	**Article Number:** 2395-0056
Author's position/theoretical position: associate professor, student, student, student,	
Keywords: SMT, NMT, Encoder, Decoder, Language Translator	

Important points, notes, quotations **Page No.**

1. . NMT systems take advantage of continuous representations that greatly ease the sparsity problem, and make use of much larger contexts, thus lessen the locality problem. **1**
2. They used common toolkits for NMT (Nematus) and traditional phrase-based statistical machine translation (Moses) with common data sets, drawn from WMT and OPUS. **2**
3. They trained their models using 8 different configurations and evaluated them using five different standard and evaluation metrics. **3**

Essential Background Information: This research paper focuses on the survey and analysis of language translation using Neural Machine Translation (NMT), which is a new approach to machine translation. It discusses different approaches, compares NMT with Statistical Machine Translation (SMT), and explores the use of encoder-decoder models in NMT

Overall argument or hypothesis: Neural Machine Translation (NMT) is a promising approach that addresses the limitations of traditional machine translation systems, offering improved performance, translation accuracy, and efficiency.

Conclusion: In practice, however, NMT systems used to be better in accuracy than phrase-based translation systems, especially when training on very large-scale datasets as used for the very best publicly available translation systems. Weaknesses of NMT are responsible for this gap: slower training and inference speed, ineffectiveness in dealing with rare words, and sometimes failure to translate all words in the source sentence.

Supporting Reasons

1. Presents empirical evidence showcasing NMT's ability to produce more accurate and fluent translations, capturing nuanced meanings and handling complex sentence structures effectively.

2. The paper highlights the advantage of end-to-end learning in NMT systems.

3. NMT models are designed to capture long-range dependencies between words in a sentence, which is often a challenge for traditional statistical models.

4. The paper discusses the ability of NMT models to adapt to new domains or languages through transfer learning.

Strengths of the line of reasoning and supporting evidence: The emergence of Transformer-based models, advancements in multilingual and low-resource NMT, improvements in attention mechanisms, and large-scale evaluation campaigns.

Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: Weaknesses of NMT are responsible for this gap: slower training and inference speed, ineffectiveness in dealing with rare words, and sometimes failure to translate all words in the source sentence.

A.7 CODE

from tkinter import * from tkinter import ttk from PIL import ImageTk, Image import csv

def interface1(): root = Tk() root.geometry('850x500+200+100') root.resizable(0,0) root.title("Instructions!!!")

background$_i mage = Image.open(r"C:4thSem_T ranslator.jpg")resized_i mage = background_i mage.resi$

bg$_i mage = ImageTk.PhotoImage(resized_i mage)$

background$_l abel = Label(root, image = bg_i mage)background_l abel.place(x = 0, y = 0, relwidth = 1, relheight = 1)$

root$_l abel = Label(text = "WelcometoLanguageTranslator", font =' Arial12bold')root_l abel.place(x = 325, y = 10)$

root$_l abel1 = Label(text = "InstructionsforLanguageTranslator", font =' Arial12bold')root_l abel1.p 300, y = 50)$

root$_l abel1 = Label(text = "1.Youcangiveinputonlyin EnglishLanguage", font =' Arial12bold')root_l ab 200, y = 100)$

root$_l abel1 = Label(text = "2.Thereareonly5languagessupportedfortranslation.", font =' Arial12bold')root_l abel1.place(x = 200, y = 150)$

root$_l abel1 = Label(text = "3.Duetolimiteddatasets, youcanonlytranslatesimplewords.", font =' Arial12bold')root_l abel1.place(x = 200, y = 200)$

root$_l abel1 = Label(text = "4.Besuretoincludepunctuationsandmakenoerrors.", font =' Arial12bold')root_l abel1.place(x = 200, y = 250)$

root$_l abel1 = Label(text = "5.Youcanalsohelpusimproveourdatasetbypressing ProvideTranslationBut Arial12bold')root_l abel1.place(x = 200, y = 300)$

root$_l abel1 = Label(text = "That'sit, YoucanusetheTranslatornow.", font =' Arial12bold')root_l abel1. 250, y = 350)$

def open$_t ranslation_i nterface()$:

root.destroy()

def back1(): root2.destroy() interface1()

def translate$_t ext()$:

error = "No translation found in the dataset" input$_t ext = Input_t ext.get("1.0","end - 1c")target_l ang = language_c odes[dest_l ang_c ombo.get()]$

$\text{if (target}_{lang} == ' es') : def load_{dataset}(file_{path}) : dataset = []with open(file_{path}, ' r', encoding = '$
$utf-8')as file : reader = csv.reader(file)for row in reader : dataset.append(row)return dataset$

$\text{def get}_{spanish_{translation}}(english_{sentence}, dataset) : for pair in dataset : if pair[0] ==$
$english_{sentence} : return pair[1]return error$

$\text{file}_{path} = r"C : 4thSem_{Translator toes.csv"} dataset = load_{dataset}(file_{path})$

$\text{input}_{sentence} = input_{text}$

$\text{spanish}_{translation} = get_{spanish_{translation}}(input_{sentence}, dataset)$

$\text{Output}_{text}.delete("1.0", END) Output_{text}.insert(END, spanish_{translation})$

$\text{elif(target}_{lang} == ' fr') : def load_{dataset}(file_{path}) : dataset = []with open(file_{path}, ' r', encoding = '$
$utf-8')as file : reader = csv.reader(file)for row in reader : dataset.append(row)return dataset$

$\text{def get}_{french_{translation}}(english_{sentence}, dataset) : for pair in dataset : if pair[0] ==$
$english_{sentence} : return pair[1]return error$

$\text{file}_{path} = r"C : 4thSem_{Translator to fr.csv"} dataset = load_{dataset}(file_{path})$

$\text{input}_{sentence} = input_{text}$

$\text{french}_{translation} = get_{french_{translation}}(input_{sentence}, dataset)$

$\text{Output}_{text}.delete("1.0", END) Output_{text}.insert(END, french_{translation})$

$\text{elif(target}_{lang} == ' de') : def load_{dataset}(file_{path}) : dataset = []with open(file_{path}, ' r', encoding = '$
$utf-8')as file : reader = csv.reader(file)for row in reader : dataset.append(row)return dataset$

$\text{def get}_{german_{translation}}(english_{sentence}, dataset) : for pair in dataset : if pair[0] ==$
$english_{sentence} : return pair[1]return error$

$\text{file}_{path} = r"C : 4thSem_{Translator to de.csv"} dataset = load_{dataset}(file_{path})$

$\text{input}_{sentence} = input_{text}$

$\text{german}_{translation} = get_{german_{translation}}(input_{sentence}, dataset)$

$\text{Output}_{text}.delete("1.0", END) Output_{text}.insert(END, german_{translation})$

$\text{elif(target}_{lang} == ' ja') : def load_{dataset}(file_{path}) : dataset = []with open(file_{path}, ' r', encoding = '$
$utf-8')as file : reader = csv.reader(file)for row in reader : dataset.append(row)return dataset$

def get$_{j}apanese_{t}ranslation(english_{s}entence, dataset) : for pair in dataset : if pair[0] == english_{s}entence : return pair[1] return error$

file$_{p}ath = r"C : 4thSem_{T}ranslator to ja.csv" dataset = load_{d}ataset(file_{p}ath)$

input$_{s}entence = input_{t}ext$

japanese$_{t}ranslation = get_{j}apanese_{t}ranslation(input_{s}entence, dataset)$

Output$_{t}ext.delete("1.0", END) Output_{t}ext.insert(END, japanese_{t}ranslation)$

else: def load$_{d}ataset(file_{p}ath) : dataset = [] with open(file_{p}ath, ' r', encoding = ' utf - 8') as file : reader = csv.reader(file) for row in reader : dataset.append(row) return dataset$

def get$_{n}epali_{t}ranslation(english_{s}entence, dataset) : for pair in dataset : if pair[0] == english_{s}entence : return pair[1] return error$

file$_{p}ath = r"C : 4thSem_{T}ranslator to ne.csv" dataset = load_{d}ataset(file_{p}ath)$

input$_{s}entence = input_{t}ext$

nepali$_{t}ranslation = get_{n}epali_{t}ranslation(input_{s}entence, dataset)$

Output$_{t}ext.delete("1.0", END) Output_{t}ext.insert(END, nepali_{t}ranslation)$

root2 = Tk() root2.geometry('850x500+200+100') root2.resizable(0, 0) root2.title('Language Translator')

background$_{i}mage2 = Image.open(r"C : 4thSem_{T}ranslator.jpg") resized_{i}mage2 = background_{i}mage2.$

bg$_{i}mage2 = ImageTk.PhotoImage(resized_{i}mage2)$

background$_{l}abel2 = Label(root2, image = bg_{i}mage2) background_{l}abel2.place(x = 0, y = 0, relwidth = 1, relheight = 1)$

title$_{l}abel = Label(root2, text = "Language Translator", font = ' Arial 20 bold') title_{l}abel.pack(pady = 20)$

language$_{c}odes = 'Spanish' : ' es', ' French' : ' fr', ' German' : ' de', ' Japanese' : ' ja', ' Nepali' : ' ne'$

Frame for input and output io$_{f}rame = Frame(root2) io_{f}rame.pack(pady = 20)$

Input Frame input$_{f}rame = Frame(io_{f}rame) input_{f}rame.grid(row = 0, column = 0, padx = 20, pady = 10, sticky = "n")$

enter$_t ext_l abel = Label(input_f rame, text = "EnterTextinEnglish:", font = "Arial12bold")enter_t ext_l ab$
10)

Input$_t ext = Text(input_f rame, width = 30, height = 10, font =' Arial12')Input_t ext.pack(pady =$
10)

Output Frame output$_f rame = Frame(io_f rame)out put_f rame.grid(row = 0, column =$
$1, padx = 10, pady = 10, sticky = "n")$

output$_l abel = Label(out put_f rame, text =' Out putText :', font =' Arial12bold')out put_l abel.pack(pad y$
10)

Output$_t ext = Text(out put_f rame, width = 30, height = 10, font =' Arial12')Out put_t ext.pack(pady =$
10)

Language Selection Frame - Output output$_l ang_f rame = Frame(out put_f rame)out put_l ang_f rame.pack()$

output$_l ang_l abel = Label(out put_l ang_f rame, text =' SelectTargetLanguage :', font ='$
$Arial12bold')out put_l ang_l abel.pack(side = LEFT, padx = 10)$

dest$_l ang_c ombo = ttk.Combobox(out put_l ang_f rame, values = list(language_c odes), width =$
$15, font =' Arial12')Reducethewidthoftheoutputlanguagecomboboxdest_l ang_c ombo.pack(padx =$
$10)dest_l ang_c ombo.set('ChooseTargetLanguage')$

Translate Button trans$_b tn = Button(root2, text =' Translate', font =' Arial12bold', pady =$
$5, command = translate_t ext, bg =' grey', fg =' white')trans_b tn.pack(pady = 10)$

back$_b tn = Button(root2, text =' Back', font =' Arial12bold', command = back1, pady =$
$5, bg =' red', fg =' white')back_b tn.place(x = 700, y = 425)$

root2.mainloop()

def provide$_t ranslation()$:

root.destroy()

def back():

root3.destroy()

interface1()

def save$_t ranslation() : input = Input_t ext3.get("1.0","end - 1c")out put = Out put_t ext3.get("1.0","end$
$1c")target_l ang = language_c odes[dest_l ang_c ombo3.get()]$

if(target$_lang == '$ es') :

file$_path = r$"$C : 4thSem_Translatortoes.csv$"$ $with open(file_path,' a', newline ='', encoding ='$ $utf-8')as file : writer = csv.writer(file)writer.writerow([input, output])$

elif(target$_lang == '$ $fr') : file_path = r$"$C : 4thSem_Translatortofr.csv$"$ $with open(file_path,' a', newline =$ $, encoding =' utf-8')as file : writer = csv.writer(file)writer.writerow([input, output])$

elif(target$_lang == '$ $de') : file_path = r$"$C : 4thSem_Translatortode.csv$"$ $with open(file_path,' a', newline =$ $, encoding =' utf-8')as file : writer = csv.writer(file)writer.writerow([input, output])$

elif(target$_lang == '$ $ja') : file_path = r$"$C : 4thSem_Translatortoja.csv$"$ $with open(file_path,' a', newline =$ $, encoding =' utf-8')as file : writer = csv.writer(file)writer.writerow([input, output])$

else: file$_path = r$"$C : 4thSem_Translatortone.csv$"$ $with open(file_path,' a', newline ='', encoding ='$ $utf-8')as file : writer = csv.writer(file)writer.writerow([input, output])$

root3 = Tk() root3.geometry('850x500+200+100') root3.resizable(0, 0) root3.title('Language Translator')

background$_image3 = Image.open(r$"$C : 4thSem_Translator.jpg$"$)resized_image3 = background_image3.$

bg$_image3 = ImageTk.PhotoImage(resized_image3)$

background$_label3 = Label(root3, image = bg_image3)background_label3.place(x = 0, y =$ $0, relwidth = 1, relheight = 1)$

title$_label3 = Label(root3, text =$ "$Language Translator$", $font =' Arial 20 bold')title_label3.pack(pady$ $20)$

language$_codes = 'Spanish' :' es','French' :' fr','German' :' de','Japanese' :' ja','Nepali' :' ne'$

Frame for input and output io$_frame3 = Frame(root3)io_frame3.pack(pady = 20)$

Input Frame input$_frame3 = Frame(io_frame3)input_frame3.grid(row = 0, column =$ $0, padx = 20, pady = 10, sticky =$ "n")

enter$_text_label3 = Label(input_frame3, text =$ "$Enter Text in English :$ ", $font =$ "$Arial 12 bold$")$ $enter_text$ $10)$

Input$_text3 = Text(input_frame3, width = 30, height = 10, font =' Arial 12')Input_text3.pack(pady =$ $10)$

Output Frame output$_frame3 = Frame(io_frame3)output_frame3.grid(row = 0, column =$

$1, padx = 10, pady = 10, sticky = "n")$

output$_label = Label(output_frame3, text =' OutputText :', font =' Arial12bold')output_label.pack(pad$
$10)$

Output$_text3 = Text(output_frame3, width = 30, height = 10, font =' Arial12')Output_text3.pack(pady =$
$10)$

Language Selection Frame - Output output$_lang_frame3 = Frame(output_frame3)output_lang_frame3.pac$

output$_lang_label3 = Label(output_lang_frame3, text =' SelectTargetLanguage :', font =$
$Arial12bold')output_lang_label3.pack(side = LEFT, padx = 10)$

dest$_lang_combo3 = ttk.Combobox(output_lang_frame3, values = list(language_codes), width =$
$15, font =' Arial12')Reducethewidthofheoutputlanguagecomboboxdest_lang_combo3.pack(padx =$
$10)dest_lang_combo3.set('ChooseTargetLanguage')$

Translate Button trans$_btn3 = Button(root3, text =' SaveTranslation', command = save_translation, font$
$Arial12bold', pady = 5, bg =' grey', fg =' white')trans_btn3.pack(pady = 10)$

back$_btn = Button(root3, text =' Back', font =' Arial12bold', command = back, pady =$
$5, bg =' red', fg =' white')back_btn.place(x = 700, y = 425)$

root3.mainloop()

login$_btn = Button(root, text =' UseLanguageTranslator', font =' Arial12bold', pady =$
$5, command = open_translation_interface, bg =' grey', fg =' white')login_btn.place(x =$
$150, y = 400)$

login$_btn2 = Button(root, text =' ProvideTranslation', font =' Arial12bold', command =$
$provide_translation, pady = 5, bg =' grey', fg =' white')login_btn2.place(x = 600, y =$
$400)$

root.mainloop()

interface1()

REFERENCES

[1] Ali Reza Ghanooni. A review of the history of translation studies. *Theory and Practice in Language Studies*, 2(1):77, 2012.

[2] Elenaz Janfaza, A Assemi, and SS Dehghan. Language, translation, and culture. In *International conference on language, medias and culture*, volume 33, pages 83–87, 2012.

[3] Manjur Kolhar and Abdalla Alameen. Artificial intelligence based language translation platform. *Intelligent Automation & Soft Computing*, 28(1), 2021.

[4] M Vaishnavi, HR Dhanush Datta, Varsha Vemuri, and L Jahnavi. Language translator application.

[5] Mahesh Vanjani and Milam Aiken. A comparison of free online machine language translators. *J. Manag. Sci. Bus. Intell*, 5:26–31, 2020.

[6] Ms Neeta Verma, Abhay Jain, Animesh Basak, and Kshitij Bharti Saksena. Survey and analysis on language translator using neural machine translation. 2018.