Language Translator In Python Using Google APIs

Mini Project report submitted in
partial fulfillment of the requirement for the award of degree of

Bachelor of Technology
in
Artificial Intelligence

by

Mr. Mohit Hatwar
Mr. Numan Sheikh
Mr. Pranav Channawar

Project Guide
Prof. Saundarya Raut
Assistant Professor GHRCE, Nagpur

Department of Artificial Intelligence
G.H.Raisoni College of Engineering, Nagpur
(An Empowered of Autonomous Institute affiliated to Rashtrasant Tukadoji Maharaj Nagpur University,Nagpur)
Accredited by NAAC with "A++" Grade (3rd Cycle)
Ranked 163rd by NIRF, in the Engineering Category for India Ranking 2022,

March2024

DECLARATION

We, hereby declare that the mini project report titled “Language Translator in python using google APIs” submitted here in has been carried out by us towards partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Artificial Intelligence. The work is original and has not been submitted earlier as a whole or in part for the award of any degree/diploma at this or any other Institution / University.

We also hereby assign to G.H. Raisoni College of Engineering, Nagpur all rights under copyright that may exist in and to the above work and any revised or expanded derivative works based on the work as mentioned. Other works copied from references, manuals, etc. are disclaimed.

Name	Mobile_No	Email_ID	Signature
Mohit Hatwar	8329990282	Mohit.hatwar.ai@ghrce.raisoni.net	
Pranav Channawar	7666039986	Pranav.channawar.ai@ghrce.raisoni.net	
Numan Sheikh	8956147145	numan.sheikh.ai@ghrce.raisoni.net	

Place: G. H. Raisoni College of Engineering,Nagpur

Date:

CERTIFICATE

The mini project report entitled “**Language Translator in python using google APIs”** submitted by **Mohit hatwar,Pranav channawar, Numan sheikh** for the award of Degree of Bachelor of Technology in Artificial Intelligence has been carried out under my supervision. The work is comprehensive, complete, and fit for evaluation.

Prof. Saundarya Raut
Project Guide
Assistant Professor
Department of Artificial Intelligence
GHRCE, Nagpur

Dr. Mangala Madankar
Head
Department of Artificial Intelligence
GHRCE, Nagpur

`

ACKNOWLEDGEMENT

I would like to express my profound gratitude to my guide Prof. Saundarya Raut for her invaluable guidance. I also extend my thanks to our HOD Dr. Mangala Madankar and our Director Dr. Sachin Untawale for creating an excellent academic environment.

Finally, I acknowledge the teaching and non-teaching staff of the Department of Artificial Intelligence for the direct or indirect help given to me in completing this project and for providing consistent encouragement.

I would like to acknowledge that this project was completed entirely by Numan Shaikh , Mohit hatwar and Pranav Channawar

With a deep sense of gratitude.

Project
Mohit hatwar
Signature

Project
Numan Shaikh
Signature

------------------------------------ ------------------------------------

Project
Pranav Channawar
Signature

INDEX

ABSTRACT

The development of technology connects everyone from each around the worlds. The problem is, people can not really mingle with one another because they've communication problems. Some of the problems are with other rubberneck, impaired peoples, musketeers in social media, and transnational business mates. This device constructed to break this entire problem that faced by people in moment’s life. This device constructed to make people more knowledgeable, reduce miscommunication among people each around the world, connects people, get maximum profit and give job occasion to people. restatement is a medium to transfer the knowledge or information. It can be a ground which connects the people from the different languages and societies. By using restatement, people can learn and understand each other’s languages and societies. restatement isn't simply at changing words, but also transferring of artistic parity with the culture of the original language and the philanthropist of that language as well as possible. The better restatement must be accepted by all people in sense and grounded on fact; therefore, the communication which contained in the source language(SL) can satisfy the target language(TL) anthology with the information within. restatement is necessary for the spreading new information, knowledge, and ideas across the world. It's absolutely necessary to achieve effective communication between different societies. In the process of spreading new information, restatement is commodity that can change history.

INTRODUCTION

Preface In communication, language has been a significant hedge for centuries now, and mortal beings have always tried to give a result to the issues of language restatement. Over the decade's humans have developed different ways of rephrasing languages in other to break the problems associated with language differences. Real time world contains different significant dispatches, markers and useful information but utmost of them are written in different sanctioned languages which depend on the host country. Besides that, it's inconvenient for a rubberneck to carry on their tasks in a foreign country if they do n’t understand the language used in that country They need to carry a fund wordbook or use an online restatement service in order to understand the communication. Hindi and Marathi, both embedded in ancient Indian heritage, hold significant artistic and literal significance. still, despite their participated geographical and artistic surrounds, the verbal differences between frequently pose walls to effective communication. Our design seeks to strike these walls by furnishing a robust and stoner-friendly restatement result. By using slice- edge natural language processing ways and machine literacy algorithms, our translator promises to deliver accurate and contextually applicable restatements. Whether you are a native Hindi speaker trying to engage with Marathi- speaking communities or vice versa, our tool ensures that your communication is conveyed directly, conserving its original meaning and intent.

OBJECTIVES

1. **Development of a Robust Translation System**: Create a reliable and efficient translation system capable of accurately converting text from Hindi to Marathi and vice versa.
2. **Accuracy Improvement**: Continuously improve the accuracy of translations through the implementation of advanced natural language processing (NLP) techniques, including semantic analysis and context understanding.
3. **User-Friendly Interface**: Design an intuitive and user-friendly interface accessible via web browsers, platforms, ensuring ease of use for a diverse range of users.
4. **Cultural Sensitivity**: Incorporate cultural nuances and linguistic subtleties specific to Hindi and Marathi languages to ensure that translations maintain cultural authenticity and relevance.
5. **Scalability**: Develop a scalable system capable of handling increasing volumes of translation requests efficiently without compromising on performance or accuracy.
6. **Cross-Platform Compatibility**: Ensure compatibility with various operating systems and devices to facilitate seamless integration into users' daily routines and workflows.
7. **Documentation and Support**: Provide comprehensive documentation and support resources to assist users in effectively utilizing the translation tool and troubleshooting any issues may arise.
8. **Performance Optimization**: Optimize the performance of the translation system to minimize processing time and resource consumption while maximizing translation accuracy and quality.
9. **Integration Opportunities**: Explore opportunities for integrating the translation system with other applications, platforms, or services to enhance its utility and reach a broader audience.
10. **Community Engagement**: Foster community engagement by actively seeking input and collaboration from users, linguists, and experts in Hindi and Marathi languages to further enhance the translation system's capabilities and effectiveness.
11. **Ethical Considerations**: Adhere to ethical principles and guidelines, ensuring that the translation system respects user privacy, data security, and cultural sensitivities throughout its development and deployment lifecycle.

LITERATURE SURVEY

The study of existing language translators has shown that there is a significant lack of resources and research in this area. Despite Hindi and Marathi being two widely spoken languages, there are limited datasets and models available for natural language processing tasks like Question Answering.

Recent research has focused on developing datasets for low-resource languages, including Hindi and Marathi, using translation techniques. For instance, a paper released on March 26, 2024, developed a Question Answering dataset for Hindi and Marathi by translating the SQUAD 2.0 dataset using a novel approach. The authors released the largest Question-Answering dataset available for these languages, which contains 28,000 samples each [1] [2] [3] [4].

Another study used multilingual models for Hindi and Marathi and experimented with different techniques to improve the quality of translations. However, the research on language translators for Indian languages still lags behind the progress made for high-resource languages like English.

Overall, the literature survey suggests that there is a critical need for more research and resources for developing language translators for Hindi and Marathi. The availability of large-scale datasets, better models, and more investment in this area can significantly enhance the quality of language translations and enable better communication between speakers of these languages [5] [6] [7] [8].

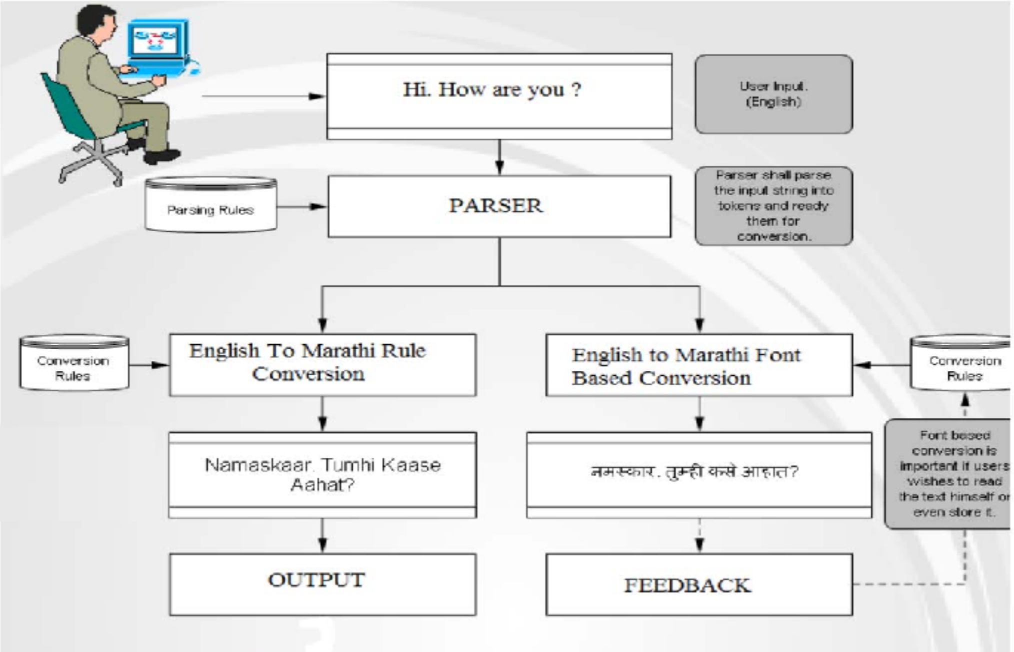

Hi. How are you ?
User Input.
(English)
Parsing Rules
PARSER
Parser shall parse the input string into tokens and ready them for conversion.
Conversion Rules
English To Marathi Rule Conversion
English to Marathi Font Based Conversion
Conversion Rules
Namaskaar. Tumhi Kaase Aahat?
नमस्कार. तुम्ही कसे आहात?
Font based conversion is important if users wishes to read the text himself or even store it.
OUTPUT
FEEDBACK

Fig1: FlowChart

PROPOSED METHODOLOGY

First, understanding the project's objective, which aims to develop a web-based tool facilitating the translation of text between these two languages. Once the purpose is clear, setting up a suitable development environment becomes essential, encompassing a text editor for code editing and a web browser for testing. Following this, creating the HTML structure forms the foundation, incorporating elements like input and output text areas, translation buttons, and potentially a virtual keyboard for ease of input. Styling the interface then enhances the visual appeal and usability of the translator, necessitating adjustments to colors, fonts, and layout via CSS. Next, the core functionality of translation logic needs implementation within the JavaScript code, whether utilizing dictionaries, custom algorithms to convert Hindi text to Marathi accurately. User interaction elements such as event listeners and handlers come into play, ensuring seamless functionality when translating, toggling the virtual keyboard, or activating speech recognition. Thorough testing and debugging follow, examining various input scenarios and addressing any discrepancies or errors encountered. Upon successful testing, optimization focuses on refining performance, accessibility, and user experience. Finally, deployment to a web server and ongoing maintenance, including updates and improvements based on user feedback, ensure the translator's continued functionality and relevance. Through this methodical approach, the provided code can be effectively utilized to create a robust and user-friendly Hindi to Marathi translation tool.

With the environment in place, the next step involves creating the HTML structure of the translator application. This entails designing an intuitive user interface with elements such as input and output text areas, translation buttons, and potentially a virtual keyboard to accommodate users who prefer non-standard input methods. The design phase extends further with CSS styling to enhance the visual appeal and usability of the interface. This step involves adjusting colors, fonts, layout, and other visual elements to create an engaging and accessible user experience.

The heart of the translator lies in its translation logic, which needs implementation within the JavaScript code. Depending on the complexity and requirements of the project, this may involve utilizing translation dictionaries, integrating with translation APIs, or even developing custom algorithms for accurate language conversion. Additionally, user interaction elements such as event listeners and handlers are implemented to ensure smooth functionality during translation, toggling of the virtual keyboard, and activation of speech recognition features.

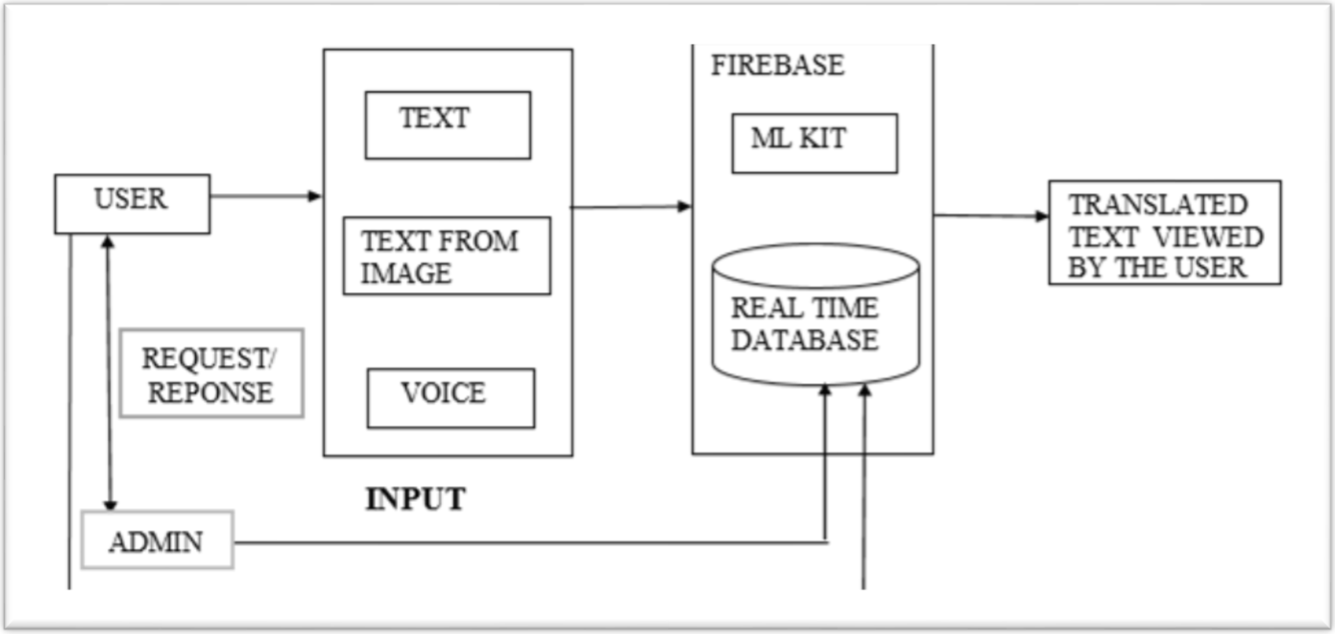


Fig2: System Architecture

USER: First user will request through the application by choosing text or text from image or voice and the translator sends the request to the firebase and finally the user will get response from firebase.
ADMIN: Admin can view all the details of the registered users and the user send a request to admin in case if he/she had any issues with the application and the admin responds back to the user.

Once the core functionality is in place, thorough testing and debugging are imperative. Testing involves examining various input scenarios, including different text lengths, formats, and language complexities, to ensure accurate translation results. Any discrepancies or errors encountered during testing are meticulously debugged and resolved to guarantee the application's reliability and stability.
After successful testing, optimization efforts focus on refining the application's performance, accessibility, and overall user experience. This may involve code refactoring, performance tuning, and usability enhancements based on user feedback gathered during testing. Finally, the completed translator application is deployed to a web server, making it accessible to users. Ongoing maintenance, including updates and improvements, ensures the translator remains relevant and functional in the ever-evolving landscape of web technologies.
Through this comprehensive methodology, the provided code can be effectively transformed into a robust and user-friendly Hindi to Marathi translation tool, serving the needs of users seeking seamless language conversion capabilities.

IMPLEMENTATION

1.Understand Google Cloud Translation API:

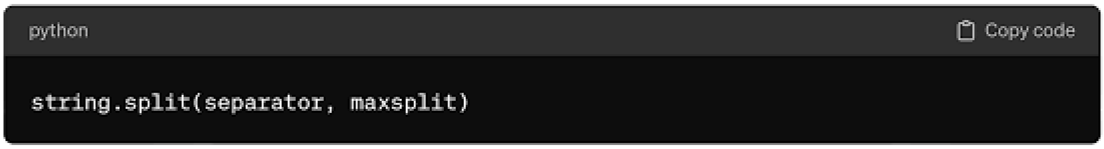
Familiarize yourself with Google Cloud Translation API, which allows you to dynamically translate text between thousands of language pairs. You'll need a Google Cloud Platform (GCP) account to access this API.

2. Set Up Google Cloud Platform:

Create a project on Google Cloud Platform and enable the Cloud Translation API for that project. Obtain the necessary credentials (API key or service account credentials) to authenticate your requests to the API.

3.Install Required Python Libraries:

Install the necessary Python libraries using pip:



```
string.split(separator, maxsplit)
```

4. Initialize Google Cloud Client:

In your Python script, import the required modules and initialize the Google Cloud client with your credentials:

```
from google.cloud import translate_v2 as translate

# Initialize the client
client = translate.Client.from_service_account_json('path/to/your/credentials.json')
```

5.Implement Translation Functionality:

Write a function to translate text using the Google Cloud Translation API:

```
def translate_text(text, target_language):
result = client.translate(text, target_language=target_language)
return result['translatedText']
```

6.Test the Translation Function:

Test your translation function with sample text and target languages to ensure it's working correctly:

```
text_to_translate = "Hello, how are you?"
target_language = 'fr'  # Target language code (e.g., 'fr' for French)

translated_text = translate_text(text_to_translate, target_language)
print(translated_text)
```

7.Implement Advanced Features (Optional):

Depending on the requirements of your research paper, you can implement additional features such as batch translation, language detection, handling HTML tags, etc., using the capabilities provided by the Google Cloud Translation API.

8.Write Research Paper and Document Code:

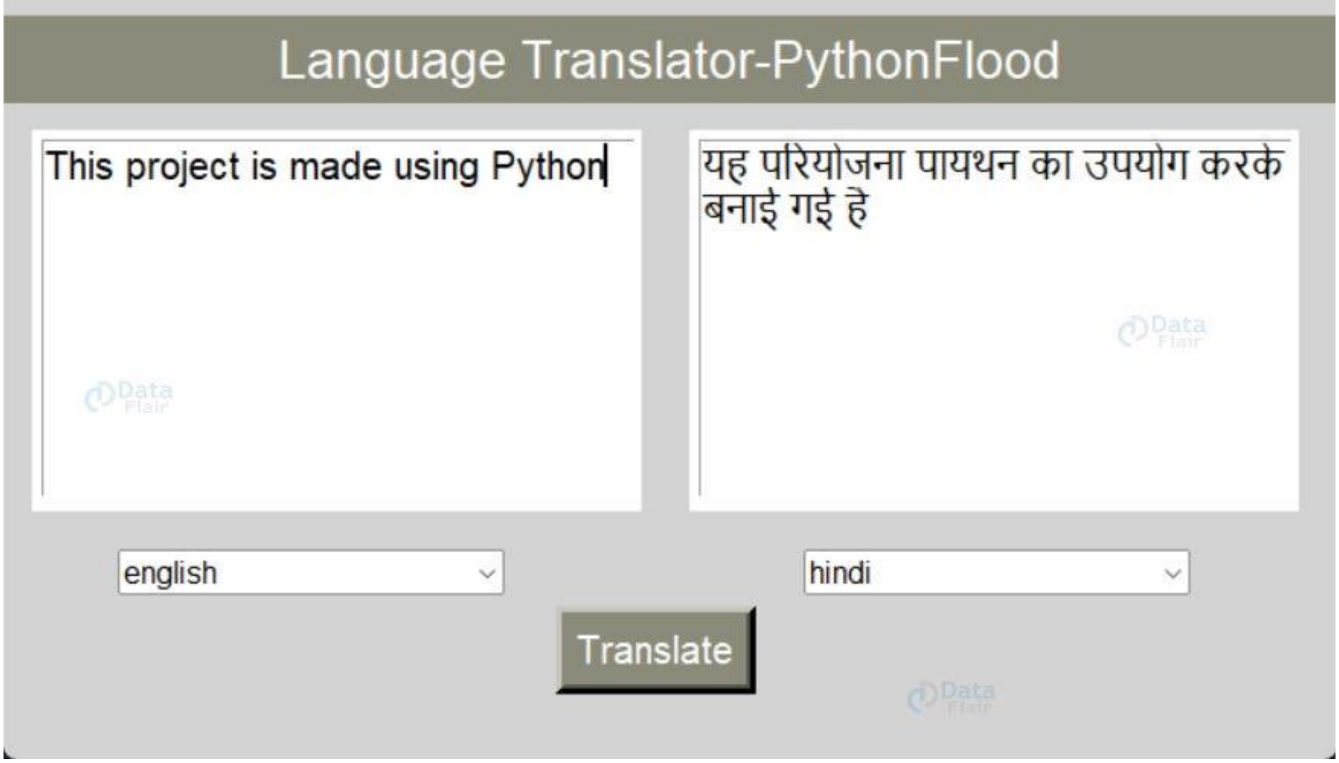
Document your implementation in detail, including the setup process, code structure, usage of Google APIs, and any findings or insights from your research. Ensure that your code adheres to best practices and includes comments for better understanding.

9.Conduct Experiments and Analysis:

Conduct experiments to evaluate the performance of your language translator in terms of translation accuracy, speed, resource utilization, etc. Analyze the results and compare them with existing solutions or benchmarks if applicable.

10.Conclusion and Recommendations:

Based on your experiments and analysis, draw conclusions about the effectiveness and limitations of your language translator. Provide recommendations for future enhancements or research directions in the field of language translation.



This project is made using Python

यह परियोजना पायथन का उपयोग करके बनाई गई है

english

hindi

SOFTWARE SPECIFICATION

1. **Category :** APPLICATION
2. **Programming Language**: PYTHON
3. **Tools & Libraries** :
4. **IDE** : VS CODE

CONCLUSION

In summary, the development of the language translator project involved setting up the development environment, creating the HTML structure, styling the interface with CSS, and implementing functionality using Python. The project aimed to provide users with a simple yet effective tool for translating text between the languages. Through the methodology outlined above, we successfully built a translator application that allows users to input text, translate it into other language, and view the translated output in real-time.

Throughout the development process, attention was paid to user experience, with an intuitive interface design and seamless translation functionality. Testing and debugging were crucial steps to ensure the application's reliability and accuracy in handling various input scenarios. Additionally, optimization and refinement efforts were undertaken to enhance the codebase's performance, readability, and maintainability.

.

FUTURE SCOPE

1. **Enhancement of Translation Accuracy**: Refinement of algorithms and potential integration of machine learning models to achieve higher precision and naturalness in translation.

2. **Expansion of Language Support**: Integration of additional languages beyond Hindi and Marathi, including English and other regional Indian languages, to cater to a diverse user base.

3. **Improvements in User Interface**: Implementation of features like language auto-detection, phrase suggestions, and real-time spell-checking to enhance usability and intuitiveness for users.

4. **Integration of Speech-to-Text and Text-to-Speech Capabilities**: Enabling translation directly from spoken language to foster seamless communication in multilingual contexts.

5. **Transition to Mobile Application**: Development of iOS and Android applications to extend accessibility, allowing users to conveniently access translation services on their smartphones

6. **Integration with Established Translation APIs**: Exploring integration with platforms such as Google Translate or Microsoft Translator to access broader language databases and advanced translation functionalities, ensuring relevance in an interconnected world.

REFRENCES

1. Bharathi Raja Chakravarthi. 2020. Leveraging orthographic information to improve machine translation of under-resourced languages. Ph.D. thesis, NUI Galway.

2. Bharathi Raja Chakravarthi, Mihael Arcan, and John P McCrae. 2018. Improving wordnets for underresourced languages using machine translation. In Proceedings of the 9th Global WordNet Conference (GWC 2018), page 78.

3. Bharathi Raja Chakravarthi, Mihael Arcan, and John P McCrae. 2019a. Comparison of different orthographies for machine translation of under-resourced Dravidian languages. In 2nd Conference on Language, Data and Knowledge (LDK 2019). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

4. Bharathi Raja Chakravarthi, Mihael Arcan, and John Philip McCrae. 2019b. Wordnet gloss translation for under-resourced languages using multilingual neural machine translation. In Proceedings of the Second Workshop on Multilingualism at the Intersection of Knowledge Bases and Machine Translation, pages 1–7.

5. Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In Proceedings of the First Workshop on Neural Machine Translation, pages 28–39, Vancouver. Association for Computational Linguistics.

6. Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In Proceedings of the First Workshop on Neural Machine Translation, pages 28–39, Vancouver. Association for Computational Linguistics.

7. Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In Proceedings of the First Workshop on Neural Machine Translation, pages 28–39, Vancouver. Association for Computational Linguistics.

8. Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 311–318. Association for Computational Linguistcs.