A

Synopsis of

"Language Translator using AI/ML"

Submitted to the D. Y. Patil University

in partial fulfillment of the requirements of the degree of

Bachelor of Technology

Computer Engineering (AI/ML)

By

Nayan Zagade : B - 77

Aditya Bhambere : B - 93

Chinmay Kale : B - 32

Under the Guidance of

Prof. Rita Kadam

School of Engineering & Technology

Department of Computer Engineering

AY 2023-24

Department of Computer Engineering

CERTIFICATE

This is to certify that the **Synopsis** entitled

"Language Translator using AI/ML"

Submitted by

Nayan Zagade : B - 77

Aditya Bhambere : B - 93

Chinmay Kale : B - 32

is an account of bona fide work carried out by him/her at the Department of Computer Engineering, in partial fulfillment of the **Bachelor of Technology Computer Engineering AI/ML, D. Y. Patil University, Pune**

Prof. Rita Kadam	**<Project Coordinator Name>**
Guide	Project Coordinator
Prof. Malayaj Kumar	**Dr. Pranav Charkha**
HOD	Dean SoET

AY 2023-24

TABLE OF CONTENTS

Chapter 1: Introduction

In today's globally interconnected world, language translation plays a pivotal role in breaking down communication barriers and facilitating cross-cultural interactions. The ability to seamlessly and accurately translate languages has profound implications for international business, education, healthcare, diplomacy, and countless other domains. As the demand for efficient language translation solutions continues to grow, the integration of artificial intelligence (AI) and machine learning technologies has emerged as a transformative force in this field.

This project represents a pioneering effort to harness the power of AI and machine learning to build an advanced language translator. Leveraging the capabilities of Python, Keras, TensorFlow, Hugging Face Transformers, Django, and SQLite, this project endeavors to create a highly capable language translation system that transcends the boundaries of traditional rule-based translation approaches.

1.1 Motivation

The motivation behind this project is driven by the limitations of existing language translation methods and the potential for AI to revolutionize this space. Traditional rule-based translation systems often struggle with context, idiomatic expressions, and language nuances, resulting in translations that may lack fluency and naturalness. In contrast, AI-based translation models have shown remarkable progress in understanding and generating human-like translations, making them ideal candidates for addressing these challenges.

The utilization of Python, Keras, TensorFlow, Hugging Face Transformers, Django, and SQLite allows for the development of a comprehensive and user-friendly language translator that can be deployed as a web application. This project aims to deliver a system that not only excels in translation accuracy but also provides a seamless user experience.

1.2 Objectives

The primary objectives of this project are as follows:

- Develop a language translation model using state-of-the-art deep learning techniques, particularly leveraging the Hugging Face Transformers library.
- Create a web-based user interface for interacting with the translation system, built using the Django web framework.
- Integrate SQLite as the backend database to store and retrieve translation history.
- Evaluate the system's translation accuracy, performance, and user-friendliness.
- Explore the advantages, disadvantages, and potential applications of the AI-powered language translator.

1.3 Scope

This project will focus on the translation of common languages, with English serving as the source language. The system will provide translation services for multiple target languages. The scope encompasses the development of the translation model, the creation of a web-based front end, and the incorporation of an SQLite database for storing translation data. The evaluation will be based on standard metrics for translation quality.

1.4 Structure of the Synopsis

This synopsis is structured as follows: Chapter 2 presents a review of past work in language translation and AI, while Chapter 3 delves into the specific problem formulation. Chapter 4 outlines the methodology, detailing the technologies and techniques employed in building the translator. Chapter 5 explores the system's advantages, disadvantages, and potential applications. Finally, Chapter 6 concludes the synopsis and suggests directions for future research.

In summary, this project seeks to harness the capabilities of Python, Keras, TensorFlow, Hugging Face Transformers, Django, and SQLite to create an innovative language translator with AI at its core, addressing the complexities and challenges of language translation in the digital age.

Chapter 2: Review of Past Work

In the realm of language translation and artificial intelligence (AI), significant strides have been made in recent years, laying the foundation for innovative projects that seek to enhance translation capabilities. This section provides an overview of the noteworthy developments, resources, and insights gathered from authoritative sources such as Academic papers, Research works, and Field experiments conducted in the realm of language translation using AI/ML.

Here is the list of reviews and summaries from various research publications:

With the rapid development of big data and deep learning, breakthroughs have been made in phonetic and textual research, the two fundamental attributes of language. Language is an essential medium of information exchange in teaching activity. The aim is to promote the transformation of the training mode and content of translation major and the application of the translation service industry in various fields. Based on previous research, the SCN-LSTM (Skip Convolutional Network and Long Short-Term Memory) translation model of deep learning neural network is constructed by learning and training the real dataset and the public PTB (Penn Treebank Dataset). The feasibility of the model's performance, translation quality, and adaptability in practical teaching is analysed to provide a theoretical basis for the research and application of the SCN-LSTM translation model in English teaching. The results show that the capability of the neural network for translation teaching is nearly one times higher than that of the traditional N-tuple translation model, and the fusion model performs much better than the single model, translation quality, and teaching effect. To be specific, the accuracy of the SCN-LSTM translation model based on deep learning neural network is 95.21%, the degree of translation confusion is reduced by 39.21% compared with that of the LSTM (Long Short-Term Memory) model, and the adaptability is 0.4 times that of the N-tuple model. With the highest level of satisfaction in practical teaching evaluation, the SCN-LSTM translation model has achieved a favourable effect on the translation teaching of the English major

From - The use of machine translation algorithm based on residual and LSTM neural network in translation teaching

Beibei Ren

Published: November 19, 2020

As we seen that AI has played an important role for translation from English to Bengali and vice versa. AI translator has huge amounts of potential when it comes to language learning, understanding the different languages and etc. Students can be benefited by it, because some students usually go to foreign or other state of their country for higher studies, so they have to face difficulties and a language barrier. Language translator using AI will be reducing their difficulties to understand their language and habituated with it. It is not only beneficial for the students but also it has global aspects. In few countries, people do not understand English, they only know their local language. So, removing this problem, it is very important to implement the language translator system.

From - Language Translation Using Artificial Intelligence

Authors: Rupayan Dirghangi, Koushik Pal, Sujoy Dutta, Arindam Roy, Rahul Bera, Manosijo Ganguly, Dipankar Pariary, Karan Kuma

From - Direct Speech to Speech Translation Using Machine Learning Sireesh Haang Limbu

Language is the unit of our whole communication system. As we all know there are many languages in our world depending on their geographical location, so to remove this language differences or language barrier, here we explain language translator system in this paper. It focuses on the translator system from English to Bengali and vice versa using machine translation. Bangla is widely spoken language and it is the fifth most spoken native language and the seventh most spoken language by the overall variety of speaker in the world. But there are only a few researches in Machine Translation (MT) for Bangla have been seen in the literature previously. Therefore, this paper represents to explain a MT system for English-Bangla and Bangla-English translation. Our focus is to help the students to feel a comfort to understand their lesson in different languages. Machine Translation is a revolutionary technology, and day by day it improves. The newest phase of machine translation is Neural Machine Translation (NMT), which is based on the development of artificial intelligence. In this paper we are proposed a type of English-Bangla language translator which based on the Encoder-Decoder model of NMT.

Authors: Rupayan Dirghangi, Koushik Pal, Sujoy Dutta, Arindam Roy, Rahul Bera, Manosijo Ganguly, Dipankar Pariary, Karan Kumar

Machine translation is an ongoing field of research from the last decades. The main aim of machine translation is to remove the language barrier. Earlier research in this field started with the direct word-to-word replacement of source language by the target language. Later on, with the advancement in computer and communication technology, there was a paradigm shift to data-driven models like statistical and neural machine translation approaches. In this paper, we have used a neural network-based deep learning technique for English to Urdu languages. Parallel corpus sizes of around 30923 sentences are used. The corpus contains sentences from English-Urdu parallel corpus, news, and sentences which are frequently used in day-to-day life. The corpus contains 542810 English tokens and 540924 Urdu tokens, and the proposed system is trained and tested using 70: 30 criteria. In order to evaluate the efficiency of the proposed system, several automatic evaluation metrics are used, and the model output is also compared with the output from Google Translator. The proposed model has an average BLEU score of 45.83.

Syed Abdul Basit Andrabi and Abdul Wahid Department of Computer Science and Information Technology, Maulana Azad National Urdu University, Hyderabad, Telangana 500032, India

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance in complex learning tasks. Currently, the mathematical language model and the neural language model still dominate research in the field of machine translation. Mathematical translation based on statistics today is the fastest but there is a lack of time accuracy. In contrast, a network-based network has high accuracy but has a very slow calculation process. In this study, comparisons between a neural network using the Recurrent Neural Network (RNN) and a mathematical-based network and an n-gram model of two French-English Machine Translation (MT) methods were developed. A quantitative analysis of both types of stress shows that the use of RNN yields a very positive effect. This paper reveals the details of the Deep Neural Network (DNN) and the concept of in-depth learning in the field of natural language processing i.e., machine translation. Now the DNN of the day plays a major role in leaning technologies. A repetitive neural network (RNN) is the best way to learn a machine. It is a combination of a recurrent neural network and a recurrent neural network (similar to the Recursive autoencoder). This paper outlines how to train a repetitive neural network for rearrangement for a source to identify. The default encoder helps to reconstruct the vectors of the target language. Therefore, powerful hardware (GPU) support is required. The GPU improves system performance by reducing training time.

Aman Sharma, Mr. Vibhor Sharma, Maharaja Agrasen Institute of Technology, Delhi Guru Gobind Singh Indraprastha University, (GGSIPU), India.

In this paper, we focus on the unidirectional translation of Kannada text to English text using Neural Machine Translation (NMT). From studies, we found that using Recurrent Neural Network (RNN) has been the most efficient way to perform machine translation. In this process we have used Sequence to Sequence (Seq2Seq) modelled dataset with the help of Encoder-Decoder Mechanism considering Long Short-Term Memory (LSTM) as RNN unit. We have compared our result concerning to Statistical Machine Translation (SMT) and obtained a better Bi-Lingual Evaluation Study (BLEU) value, with an accuracy of 86.32%.

Author - Pushpalatha Kadavigere Nagaraj* **|** Kshamitha Shobha Ravikumar **|** Mydugolam Sreenivas Kasyap **|** Medhini Hullumakki Srinivas Murthy **|** Jithin Paul

Objectives: To develop a Neural Machine Translator which can be integrated to the chatting applications which will be helpful for the users who are convenient with their regional languages.
Methods: Neural machine translation is an approach in machine translation which uses an artificial neural network to predict the likelihood of a sequence of words. It is typically modeling entire sentences in a single integrated model. NMT provides more accurate translation by taking into account the context in which a word is used, rather than just translating each individual word on its own.
Findings: We used LSTM to build our model and we were able to get the Hindi sentences for the corresponding English sentences which contains the words count less than are equal to 5 accurately. We were getting translation for sentences more than 5 words also but not all. Like if we test for 100 sentences having more than 5 words, we got almost 75 to 80 sentences accurately.

Arun P N, Harikesh B G, Rakesh S R ,Sahithya A N, Jayanna H S, N Siddaganga Institute of Technology (SIT), Tumkuru- 572103, Karnataka

Chapter 3: Problem Formulation

Language barriers have long been a hindrance to effective communication and understanding in our increasingly interconnected world. People from diverse linguistic backgrounds often face challenges in comprehending foreign languages, leading to miscommunication, misunderstandings, and missed opportunities for meaningful interactions. In this era of globalization, where cross-border collaboration and multicultural exchanges are the norm, the need for a solution to bridge the communication gap and extract contextual meaning from foreign languages has never been more pressing.

3.1 The Challenge

The challenge at hand is twofold:

1. **Language Comprehension:** Many individuals encounter difficulties in understanding spoken or written content in languages that are not their native tongue. This impedes their ability to access information, engage in discussions, or participate in activities that require multilingual competence.
2. **Contextual Understanding:** Language is not just a sequence of words; it is a carrier of culture, nuance, and context. Accurate translation goes beyond mere word substitution; it involves capturing the underlying meaning and cultural connotations to ensure effective communication.

3.2 Leveraging AI to Address the Challenge

The proposed solution to these challenges is the utilization of Artificial Intelligence (AI) and machine learning techniques to develop an advanced language translator. By leveraging AI, we aim to not only facilitate language comprehension but also preserve and convey the contextual richness of the source content.

3.3 Objectives

The objectives of this project, driven by the identified problem, are as follows:

1. **Efficient Language Translation:** Create an AI-powered language translation system capable of translating text or speech from one language to another accurately and rapidly.
2. **Contextual Understanding:** Develop the ability to capture and preserve the context, tone, and cultural nuances of the source content to ensure that translations are not only linguistically accurate but also contextually relevant.
3. **User-Friendly Interface:** Design a user-friendly web-based interface using Django, which enables users to interact with the translation system effortlessly.
4. **Translation History:** Implement a database using SQLite to store and retrieve translation history, allowing users to access their previous translations.
5. **Evaluation:** Assess the translation system's performance using standard evaluation metrics, including translation quality, speed, and user satisfaction.

3.4 Scope

The scope of this project encompasses the creation of a comprehensive AI-driven language translator that assists users in understanding and communicating in foreign languages. The focus is on common languages with English as the source language and support for multiple target languages. The project includes the development of both the translation model and the user interface while integrating a database for translation history.

3.5 Significance

Addressing the challenges of language comprehension and contextual understanding through AI-driven language translation has significant implications for various domains, including international business, education, diplomacy, healthcare, and cross-cultural interactions. The successful development of an efficient and context-aware language translator can foster effective global communication and promote cultural understanding.

The subsequent chapters will delve into the methodology employed to achieve these objectives, explore the system's advantages, disadvantages, and practical applications, and ultimately draw conclusions about the project's significance and future directions.

Chapter 4: Methodology

The development of an advanced language translator with a focus on language comprehension and contextual understanding entails a methodical approach that combines the strengths of Recurrent Neural Network (RNN) models, Keras, TensorFlow, and Python. This section outlines the key steps and methodologies that will be employed in the project.

4.1 Data Collection and Preprocessing

The success of any machine learning-based language translation system hinges on the quality and diversity of the training data. The following steps will be taken:

- **Data Acquisition:** Curate a diverse and extensive dataset comprising text and audio content in the source (English) and target languages.
- **Data Preprocessing:** Clean and preprocess the dataset, including tokenization, punctuation removal, and special character handling. For audio data, audio-to-text conversion will be performed.

4.2 Model Selection and Training

The project will leverage RNN-based models, in particular, Long Short-Term Memory (LSTM) networks, which are well-suited for sequential data tasks like language translation. The steps involved in model selection and training are as follows:

- **Model Architecture:** Design and implement RNN-based model architectures using Keras. Configure the models for sequence-to-sequence tasks.
- **Transfer Learning:** Utilize pre-trained transformer models from the Hugging Face Transformers library to enhance translation quality and efficiency.
- **Training:** Train the models using the pre-processed dataset, optimizing for translation quality and contextual understanding.

4.3 User Interface Development

To facilitate user interaction and seamless translation, a web-based user interface will be created using Django:

- **Frontend Design:** Design an intuitive and responsive user interface for entering source text, selecting target languages, and displaying translations.
- **Backend Integration:** Integrate the trained models into the Django backend to enable real-time translation requests.

4.4 Evaluation and Validation

The project's success will be evaluated based on several criteria:

- **Translation Quality:** Employ standard metrics such as BLEU (Bilingual Evaluation Understudy) and METEOR to assess translation quality and fluency.
- **Speed:** Measure the translation speed to ensure rapid response times.
- **User Satisfaction:** Conduct user testing and feedback collection to gauge user satisfaction and identify areas for improvement.

4.5 Deployment and Testing

The final stage of the project involves deploying the language translator on a web server and conducting comprehensive testing:

- **Deployment:** Host the translation system on a web server to make it accessible to users.
- **Testing:** Perform extensive testing, including unit testing, integration testing, and usability testing to identify and rectify any issues.

4.6 Documentation and Future Directions

Throughout the project, thorough documentation will be maintained to facilitate reproducibility and future enhancements. The project's potential future directions, such as multilingual support and integration with other platforms, will also be explored.

The methodologies outlined in this chapter provide a structured approach to the development of the AI-powered language translator, incorporating RNN models via Hugging Face, Keras, TensorFlow, and Python. These methodologies aim to deliver a robust and context-aware translation system that addresses the challenges of language comprehension and contextual understanding. The subsequent chapters will explore the system's advantages, disadvantages, and practical applications, ultimately concluding its significance and future directions.

Chapter 5: Advantages, Disadvantages, and Applications

The development of an AI-powered language translator using RNN models via Hugging Face, Keras, TensorFlow, and Python introduces a range of advantages, disadvantages, and diverse applications.

5.1 Advantages

5.1.1 Enhanced Language Comprehension

- The AI-powered translator enhances language comprehension by providing accurate translations in real-time, enabling users to understand foreign languages more effectively.

5.1.2 Contextual Understanding

- The system preserves contextual nuances, cultural references, and idiomatic expressions, ensuring that translations are not only linguistically accurate but also contextually relevant.

5.1.3 Accessibility

- The web-based interface allows users to access the translation system from anywhere with an internet connection, promoting inclusivity and accessibility.

5.1.4 Rapid Response

- RNN-based models, coupled with optimized deployment, ensure rapid translation responses, making it suitable for real-time conversations and interactions.

5.1.5 User-Friendly

- The user-friendly interface simplifies the translation process, making it accessible to individuals with varying levels of technical expertise.

5.1.6 Translation History

- The integration of a database facilitates the retrieval of previous translations, improving user convenience.

5.2 Disadvantages

5.2.1 Language Limitations

- The system may have limitations in handling less common languages or dialects due to the availability of training data.

5.2.2 Computational Resources

- Training and deploying AI models can be resource-intensive, potentially limiting accessibility on low-powered devices.

5.2.3 Translation Errors

- While AI-based translation systems have improved significantly, they may still occasionally produce errors or misinterpretations, especially in complex or specialized content.

5.3 Applications

5.3.1 International Business

- The language translator facilitates cross-border business communication, enabling companies to effectively engage with international clients, partners, and customers.

5.3.2 Education

- In educational settings, the system can aid students in understanding foreign language content, enhancing language learning and cross-cultural understanding.

5.3.3 Healthcare

- Medical professionals can use the translator to bridge language gaps when communicating with patients from diverse linguistic backgrounds, improving the quality of healthcare services.

5.3.4 Diplomacy and International Relations

- The translator can assist diplomats and international organizations in diplomatic negotiations and communications, fostering collaboration on a global scale.

5.3.5 Tourism and Travel

- Travelers can rely on the system to navigate foreign destinations, access local information, and communicate with residents.

5.3.6 Content Localization

- Businesses and content creators can use the translator to adapt and localize their content for global audiences, expanding their reach.

5.3.7 Cross-Cultural Communication

- The system promotes cross-cultural communication and understanding, breaking language barriers and fostering a sense of global interconnectedness.

5.3.8 Emergency Situations

- In emergencies, the translator can aid first responders and humanitarian organizations in communicating with affected populations, ensuring effective disaster relief efforts.

In conclusion, the AI-powered language translator project offers numerous advantages, including enhanced language comprehension, accessibility, and rapid response. However, it also has limitations, such as language constraints and occasional errors. Its wide-ranging applications span industries such as international business, education, healthcare, diplomacy, tourism, content localization, and emergency response, underscoring its potential to make a significant impact on global communication and cooperation. The following chapter will conclude and outline future directions for this transformative project.

Chapter 6: Conclusion

The development of an AI-powered language translator using Recurrent Neural Network (RNN) models via Hugging Face, Keras, TensorFlow, and Python represents a significant stride toward addressing the challenges of language comprehension and contextual understanding in our interconnected world. This transformative project has yielded valuable insights and outcomes that warrant reflection and consideration.

6.1 Achievements

Throughout this project, several key achievements have been realized:

- **Enhanced Language Comprehension:** The AI-powered translator has demonstrated the capability to significantly enhance language comprehension, making foreign languages more accessible and understandable to a wide audience.
- **Contextual Understanding:** By preserving context, cultural nuances, and idiomatic expressions, the system has surpassed the boundaries of conventional translation approaches, ensuring that translations are contextually relevant and culturally sensitive.
- **User-Friendly Interface:** The web-based interface has simplified the translation process, providing an accessible and intuitive platform for users from diverse backgrounds.
- **Translation History:** The integration of a database has enabled users to access their translation history, enhancing user convenience and facilitating the retrieval of previous translations.

6.2 Limitations and Challenges

Despite its achievements, the project is not without its limitations and challenges:

- **Language Constraints:** The system's effectiveness may vary with less common languages or dialects due to limitations in training data availability.
- **Resource Intensiveness:** Training and deploying AI models can be resource-intensive, potentially limiting accessibility on low-powered devices.
- **Occasional Errors:** While the system has demonstrated impressive accuracy, occasional errors or misinterpretations may still occur, particularly in complex or specialized content.

6.3 Future Directions

The success of this project opens the door to several exciting future directions:

- **Multilingual Support:** Expanding the system to support a broader array of languages and dialects to serve a more diverse user base.

- **Integration with Voice Recognition:** Incorporating voice recognition capabilities to enable speech-to-text and text-to-speech translation.
- **Continuous Learning:** Implementing mechanisms for the system to continuously learn from user interactions, improving translation quality over time.
- **Cross-Platform Integration:** Exploring integration with various platforms, including mobile applications and voice assistants.
- **AI Ethical Considerations:** Addressing ethical considerations, including privacy, bias, and cultural sensitivity in AI-based language translation.

6.4 Final Thoughts

In conclusion, the AI-powered language translator project represents a significant advancement in the realm of language comprehension and contextual understanding. It has the potential to bridge communication gaps, facilitate cross-cultural interactions, and empower individuals and organizations to engage more effectively in our globalized world.

As technology continues to evolve and our understanding of language and culture deepens, the possibilities for further innovation in language translation are limitless. This project is not just an endpoint but a stepping stone towards a future where language is no longer a barrier but a bridge, fostering global connectivity, mutual understanding, and cross-cultural collaboration.

The journey continues, as we look forward to a world where language is a tool for unity and shared knowledge, transcending boundaries and bringing people closer together.

References -

https://uu.diva-portal.org/smash/get/diva2:1540841/FULLTEXT01.pdf - 1st

https://www.ijraset.com/research-paper/language-translation-using-artificial-intelligence - 2nd

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0240663 - 3rd

https://www.researchgate.net/publication/347066090_The_use_of_machine_translation_algorithm_based_on_residual_and_LSTM_neural_network_in_translation_teaching - 4th

https://www.irjmets.com/uploadedfiles/paper/volume3/issue_6_june_2021/12649/1628083502.pdf -5th

Appendix -

1. Data Set – https://www.kaggle.com/datasets/devicharith/language-translation-englishfrench

2. Code snippets –

Creation of Tokenizer

```
def create_tokenizer(lines):

    # fit a tokenizer

    tokenizer = Tokenizer()

    tokenizer.fit_on_texts(lines)

    return tokenizer



def max_len(lines):

    # max sentence length

    return max(len(line.split()) for line in lines)



def encode_sequences(tokenizer, length, lines):
```

```
    # encode and pad sequences

    X = tokenizer.texts_to_sequences(lines) # integer encode sequences

    X = pad_sequences(X, maxlen=length, padding='post') # pad sequences with 0 values

    return X


def encode_output(sequences, vocab_size):

    # one hot encode target sequence

    ylist = list()

    for sequence in sequences:

        encoded = to_categorical(sequence, num_classes=vocab_size)

        ylist.append(encoded)

    y = np.array(ylist)

    y = y.reshape(sequences.shape[0], sequences.shape[1], vocab_size)

    return y
```

Model Creation -

```
def create_model(src_vocab, tar_vocab, src_timesteps, tar_timesteps, n_units):

    # Create the model

    model = Sequential()

    model.add(Embedding(src_vocab, n_units, input_length=src_timesteps, mask_zero=True))

    model.add(LSTM(n_units))

    model.add(RepeatVector(tar_timesteps))

    model.add(LSTM(n_units, return_sequences=True))

    model.add(TimeDistributed(Dense(tar_vocab, activation='softmax')))

    return model


# Create model

model = create_model(src_vocab_size, tar_vocab_size, src_length, tar_length, 256)

model.compile(optimizer='adam', loss='categorical_crossentropy')
```

```
    history = model.fit(trainX,

              trainY,

              epochs=50,

              batch_size=64,

              validation_split=0.1,

              verbose=1,

              callbacks=[

                         EarlyStopping(

                         monitor='val_loss',

                         patience=10,

                         restore_best_weights=True

                      )

                ])
```

BELU test -

```
                                                                def bleu_score(model, tokenizer,
sources, raw_dataset):

    # Get the bleu score of a model

    actual, predicted = [], []

    for i, source in enumerate(sources):

        # translate encoded source text

        source = source.reshape((1, source.shape[0]))

        translation = predict_seq(model, tar_tokenizer, source)

        raw_target, raw_src = raw_dataset[i]

        actual.append([raw_target.split()])

        predicted.append(translation.split())


    bleu_dic = {}

    bleu_dic['1-grams'] = corpus_bleu(actual, predicted, weights=(1.0, 0, 0, 0))

    bleu_dic['1-2-grams'] = corpus_bleu(actual, predicted, weights=(0.5, 0.5, 0, 0))

    bleu_dic['1-3-grams'] = corpus_bleu(actual, predicted, weights=(0.3, 0.3, 0.3, 0))
```

```
    bleu_dic['1-4-grams'] = corpus_bleu(actual, predicted, weights=(0.25, 0.25, 0.25, 0.25))



    return bleu_dic



# Compute the BLEU Score

bleu_train = bleu_score(model, tar_tokenizer, trainX, train)

bleu_test = bleu_score(model, tar_tokenizer, testX, test)
```