IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A FORMAL LANGUAGE TRANSLATOR (TELUGU - ENGLISH)

Mrs. P. Swaroopa[1], Kalakonda Vishnu [2], Pulipati Akshay[3], S. Siva Sai Sandip[4], Vennam Vivek[5]

[1] Assistant Professor, Department of Computer Science and Engineering, ACE Engineering College, Hyderabad, Telangana, India

[2,3,4,5] IV BTech Students, Department of Computer Science and Engineering, ACE Engineering College, Hyderabad, Telangana, India

1. ABSTRACT

Language, being the carrier of information, holds utmost importance for human communication and information sharing. Difficulties arise when individuals from different language backgrounds try to interact, leading to language barriers. To address this issue, the need for efficient and rapid translators becomes evident. At first, human translators played this function, but they are not a lasting answer. Consequently, scholars have carried out numerous tests and attempts, leading to the emergence of machine translation technology. This technology has been widely utilized, exploring diverse languages, and employing various methodologies like Rule-Based, Statistical-Based, and more recently, Neural Machine Translation. Machine Translation has been a familiar notion for the last five decades, falling within the domain of Natural Language Processing (NLP), a subfield of Machine Learning. The effectiveness of machine translation heavily relies on the quality of the datasets and the training the machine has undergone. However, many challenges have emerged due to the lack of adequate datasets for certain languages, often referred to as low-resource languages.

Our project primarily focuses on translating sentences, poems, and proverbs from South East Indian Languages, specifically Telugu, into English. As previously mentioned, Telugu is considered a low-resource language and is notoriously challenging for human learners. It boasts a wide array of proverbs and poems, but unlike English and other languages, Telugu has not been extensively translated. Previous attempts to translate Telugu poems and proverbs have yielded unsatisfactory results. Our main objective, therefore, is to develop a machine translation model using OpenNMT that can accurately translate Telugu sentences, poems, and proverbs into English.

2. INTRODUCTION

India is a land of incredible diversity, and this diversity extends to its languages. With over 1.3 billion people, India is home to thousands of languages and dialects. The Constitution of India recognizes 22 officially recognized languages. Additionally, there are numerous regional languages and dialects spoken across different states and communities. This linguistic diversity brings with it a unique challenge: language barriers. While many Indians are bilingual or multilingual, there are still instances where effective communication becomes hindered due to language differences.

To surmount these linguistic obstacles, translators assume a vital function in enabling proficient communication. Human translators act as intermediaries, bridging the divide between individuals who possess distinct linguistic abilities. However, with such a vast linguistic landscape in India, the demand for translators often surpasses the available supply, making it challenging to meet the needs of all language pairs. In recent years, advancements in technology have offered a promising solution to this problem. The utilization of machine translation, propelled by the advancements in Natural Language Processing (NLP), has emerged as a valuable resource in dismantling language obstacles. NLP, which falls under the domain of artificial intelligence, concentrates on empowering computers to comprehend, interpret, and generate human language.

Machine translators harness the potential of NLP algorithms to process and convert text or speech from one language to another. They analyse the grammar, syntax, and semantics of the source language and generate an output that conveys the intended meaning in the target language. By automating the translation process, machine translators offer a scalable and efficient solution to overcome language barriers.

Indeed, while machine translators have made significant advancements in recent years, they still face limitations when it comes to translating certain forms of linguistic expression, such as poems and proverbs. When it comes to translating from Telugu to English, machine translators still face challenges, particularly with regard to translating poetry and proverbs. These forms of expression often rely on cultural references, figurative language, and context-specific meanings that may not have direct equivalents in the target language. Translating Telugu poems into English requires not only a grasp of the literal meaning but also an understanding of the poetic devices, rhythm, and cultural nuances specific to Telugu literature. Machine translators, which primarily operate on rule-based or statistical algorithms, struggle to capture the intricacies of Telugu poetry and convey them effectively in English.

3. EXISTING METHODS

Translation models are computational systems designed to convert text or speech from one language to another. These models utilize various approaches and algorithms to achieve accurate and fluent translations. Over the years, different types of translation models have been developed, each with its own unique characteristics and capabilities. Some of the Translation models are.

Rule-Based Translation Models:

Rule-based translation models rely on linguistic rules created by human experts. These rules define the grammar, syntax, and semantic relationships between words and phrases in both the source and target languages. The translation process involves applying these rules systematically to generate the translated output. Rule-based models offer fine-grained control over translation quality and can handle complex linguistic phenomena. However, they require manual rule development and may struggle with ambiguity and idiomatic expressions. These are some of the merits and demerits of the Rule-Based Translation Models.

Merits:

1. Explicit rules allow for fine-grained control over translation quality.
2. Rule-based systems can handle complex linguistic phenomena and domain-specific terminology effectively.
3. They are more interpretable, making it easier to understand and modify the translation rules.

Demerits:

1. Developing and maintaining linguistic rules can be time-consuming and require expert knowledge.
2. Rule-based systems often struggle with handling ambiguity and idiomatic expressions.
3. They may lack coverage for rare or evolving linguistic patterns and vocabulary.

Statistical Translation Models:

Statistical translation models make use of extensive parallel corpora, which consist of aligned texts in both the source and target languages. These models employ statistical algorithms to acquire patterns and probabilities from the parallel data. By examining the occurrences of word or phrase translations, statistical models generate translations based on the most probable matches found in the training data. Statistical models are adaptable to different language pairs and can handle idiomatic expressions, but they require substantial amounts of training data and may produce suboptimal translations in data-scarce scenarios.

Merits:

1. Statistical systems can handle a wide range of language pairs and adapt to new languages with sufficient training data.
2. They can capture context and word associations effectively through statistical analysis.
3. Statistical models can handle idiomatic expressions and rare linguistic patterns.

Demerits:

1. Training data requirements can be extensive, making it challenging for low-resource languages.
2. Statistical models may produce suboptimal translations in cases where the training data is insufficient or unrepresentative.
3. They may struggle with translating rare or unseen words and phrases accurately.

Neural Machine Translation (NMT) Models:

Neural Machine Translation (NMT) represents a methodology for machine translation that harnesses artificial neural networks, particularly recurrent neural networks (RNNs) or transformer models, to acquire knowledge and produce translations. NMT models have gained significant popularity and have shown remarkable improvements in translation quality compared to previous machine translation approaches. Here are the key components and workings of Neural Machine Translation:

1. Encoder-Decoder Architecture:

   NMT models typically employ an encoder-decoder architecture. The encoder reads and encodes the source sentence into a fixed-length vector representation called the "thought vector" or "context vector." The decoder then takes this vector as input and generates the corresponding translated output.

2. Word Embeddings:

Word embeddings are a fundamental component of NMT models. They represent words as dense, continuous vectors in a high-dimensional space. Word embeddings capture semantic relationships and contextual information, allowing the model to learn meaningful representations of words and their associations.

3. Encoder:

The encoder receives the source sentence as a sequence of word embeddings and processes it sequentially. In recurrent-based NMT models, such as the long short-term memory (LSTM) or gated recurrent unit (GRU), the encoder's recurrent units analyze the input words one by one, updating the hidden state at each step. The hidden state captures contextual information from the preceding words. The final hidden state represents the entire source sentence, summarizing its meaning and context.

4. Attention Mechanism:

Attention is a key feature of NMT models that allows the model to focus on different parts of the source sentence during translation. The attention mechanism computes weights that indicate the relevance of each word in the source sentence to the generation of a particular word in the target sentence. This attention mechanism helps the model align words between the source and target languages, enabling it to handle long-range dependencies and improve translation quality.

5. Decoder:

The decoder takes the context vector from the encoder and generates the translated output. Like the encoder, it may employ recurrent-based units or transformer models. At each step, the decoder predicts the most likely target word based on the context vector, the previously generated words, and the target language model. The predicted word is then fed back into the decoder to generate the subsequent word, and this process continues until the end of the sentence or a predefined termination condition is met.

6. Training and Optimization:

NMT models undergo training using extensive parallel corpora, comprising aligned sentences in both the source and target languages. Throughout the training process, the model's parameters are fine-tuned to minimize the disparity between the predicted translations and the reference translations present in the training data. This is achieved through techniques like backpropagation and gradient descent, where the model learns to adjust the weights of its neural network layers to improve translation accuracy.

Merits of NMT:

1. NMT models have shown significant improvements in translation quality, capturing complex sentence structures and handling long-range dependencies more effectively.
2. They can handle a wide range of linguistic phenomena, including idiomatic expressions and rare vocabulary.
3. NMT models have the ability to learn contextual representations, enabling them to produce more fluent and natural-sounding translations.

Demerits of NMT:

1. NMT models require substantial amounts of training data to achieve optimal performance, making them sensitive to data scarcity for certain language pairs.
2. They can be computationally intensive and require powerful hardware resources for training and inference.
3. NMT models may produce translations that are overly fluent but lack accuracy or fail to capture specific source language nuances.

Each translation model type has its own merits and demerits, and their selection depends on various factors, including the available resources, language pair, domain, and translation quality requirements. Ongoing research and development continue to explore ways to enhance the capabilities of translation models and improve translation accuracy across different languages and domains.

4. PROPOSED METHOD USING OPENNMT

Our Purpose:

The main motivation behind building a Telugu to English translator is the inherent challenge faced by existing translators in accurately translating Telugu poems and proverbs into English. Language translation, especially for creative and poetic expressions, poses unique difficulties that conventional translation systems struggle to overcome. Poems and proverbs often contain cultural nuances, figurative language, wordplay, and deep-rooted contextual meanings that are hard to capture with a literal translation approach. The rich literary heritage of Telugu, with its extensive collection of poems and proverbs, demands a specialized translation solution that can accurately convey the essence and intended meaning of these artistic expressions in English. By developing a dedicated Telugu to English translator specifically designed to handle the intricacies of poetic and proverbial language, we aim to bridge the gap and provide a more comprehensive and faithful translation experience for these culturally significant forms of expression.

OpenNMT:

OpenNMT is a popular open-source toolkit utilized for Neural Machine Translation (NMT) tasks. It offers a wide range of tools and functionalities that facilitate the creation, training, and deployment of NMT models effortlessly. With its customizable and modular architecture, OpenNMT allows users to adapt and configure various components, including tokenization, encoding, decoding, and attention mechanisms, to meet their specific needs. The toolkit supports advanced NMT architectures such as LSTM and transformers, ensuring the production of high-quality translations. OpenNMT excels in efficient training and inference, capitalizing on parallel processing techniques and optimizations for both CPUs and GPUs. Its pre-processing utilities assist in data preparation, while the strong community support encourages collaboration and knowledge exchange. OpenNMT serves as a valuable resource for researchers, developers, and practitioners in the NMT domain, empowering them to explore novel ideas, experiment with diverse techniques, and achieve accurate and fluent translations across multiple languages.

OpenNMT has gained popularity in the NMT research community due to its flexibility, modularity, and ease of use. It has been utilized in various research projects, commercial applications, and academic studies, contributing to advancements in the field of NMT. The active development and community support ensure that OpenNMT remains up-to-date with the latest research and technological advancements in the NMT domain.

Our Model Preparation:

Our motive as stated above was simple, we had to create a Translation model which can translate Telugu sentences, Poems and Proverbs into English. We have opted OpenNMT to build our model, as it is easy to use and flexible it made our work easier, we have followed the following steps to build our model.

1. **Data Preparation:**

   The first step involves around collecting the data. We have prepared our data by collecting parallel sentences in the source i.e, Telugu, and target i.e, English languages. We ensured that the data is properly aligned, with corresponding sentences or texts in each language and organized the data into separate source and target files. As we were on shortage of dataset which contain Telugu poems and meanings.

2. **Data Pre-processing:**

   Preprocess the training data to convert it into a suitable format for training. This typically involves tasks such as tokenization, normalization, and possibly sub word segmentation. OpenNMT provides preprocessing utilities that can be used to prepare the data. We need to specify the preprocessing options in a configuration file.

3. **Model Configuration:**

   Create a configuration file that specifies the architecture and hyperparameters of the NMT model. OpenNMT supports various model architectures, such as LSTM or transformers. You can customize the configuration file to adjust parameters like the number of layers, hidden sizes, dropout rates, and more.

   The “config.yaml” file is a crucial configuration file that defines the settings and parameters for training and running your NMT model. It provides a structured way to specify various components and options required for the model's behaviour. we find sections like Data configuration, vocabulary configuration, model configuration, etc. By modifying and customizing the options within this file, you can fine-tune the behaviour of your NMT model according to your specific requirements. The “config.yaml” file acts as a central configuration hub, allowing you to experiment with different settings and easily reproduce your model's behaviour.

4. Vocabulary Generation:

Generate vocabulary files for both the source and target languages. These vocabulary files contain a mapping of words (or sub words) to unique numerical indices. OpenNMT provides a tool to generate the vocabularies based on the pre-processed training data.

The command “!onmt_build_vocab -config drive/MyDrive/Data/config.yaml -n_sample 100000” is used in OpenNMT to build vocabulary files based on a specified configuration file. By running this command, OpenNMT will read the specified configuration file, extract the necessary information such as the paths to the source and target training data, and build vocabulary files accordingly. The resulting vocabulary files will contain a mapping of words (or subwords) to unique numerical indices, which will be used during the training and inference stages of the NMT model.

5. Training:

Start the training process by running the OpenNMT training command, specifying the paths to the pre-processed training data, vocabulary files, and configuration file. OpenNMT will initialize the model with the specified architecture and hyperparameters and begin training on the provided data. During training, the model's parameters are updated iteratively to minimize the difference between predicted translations and the reference translations.

The command "!onmt_train -config drive/MyDrive/Data/config.yaml --model_config drive/MyDrive/Data/config.yaml" is used in OpenNMT to initiate the training process for a neural machine translation (NMT) model. By executing this command, OpenNMT will read the main configuration file and the model configuration file to obtain all the necessary information for the training process. It will initialize the NMT model based on the provided architecture settings and begins training using the training data specified in the configuration file.

During training, OpenNMT will iterate through the training data, optimizing the model's parameters to minimize the difference between the predicted translations and the reference translations. It will follow the specified training parameters, such as the batch size, optimizer type, learning rate, and gradient clipping, to update the model weights and biases. The training process continues for the specified number of steps or epochs as defined in the configuration file. OpenNMT will periodically evaluate the model's performance on the validation data using the specified evaluation metrics, allowing you to monitor the progress of the training.

6. Evaluation:

Periodically evaluate the model's performance using a validation dataset. OpenNMT allows you to generate translations from the model and calculate evaluation metrics such as BLEU score, which measures the similarity between the model's translations and the reference translations. Evaluation helps monitor the progress of the training and allows you to fine-tune the model if necessary.

7. Deploying the model:

Deploying a model refers to making a trained model available for practical use, typically in a production environment or as a part of an application. It involves integrating the model into a system or framework that allows users to access its functionality and make predictions or generate translations.

5. PERFORMANCE EVALUATION

Integration and Experimental Results:

To evaluate the performance and effectiveness of the translator project, a series of experiments were conducted. The experiments focused on assessing the accuracy of the translations and the overall user experience. The following are some of the experimental results obtained.

SNO	INPUT	OUTPUT
1.	చెప్పులోన రాయి చెవిలోన జోరీగ కంటిలోన నలుసు కాలి ముల్లు ఇంటిలోన పోరు ఇంతింత గాదయ విశ్వధాభిరామ, వినుర వేమ	A stone in shoe, a fly near ear Pollen in eye, a thorn in foot A quarrel in the house, are unbearable Beloved of the Bounteous, Vema, listen!
2.	ఉప్పు కప్పురంబు నొక్క పోలికనుండు చూడ చూడ రుచులు జాడ వేరు ఫురుషులందు ఫుణ్య పురుషులు వేరయ విశ్వధాభిరామ, వినుర వేమ	Salt and camphor look similar, closer observation shows their taste is different Among men, virtuous people stand apart Beloved of the Bounteous, Vema, listen!
3.	గంగి గోవు పాలు గరిటడైనను చాలు కడివెడైననేమి ఖరము పాలు భక్తి కలుగు కూడు పట్టెడైనను చాలు విశ్వధాభిరామ, వినుర వేమ	A ladleful of a Sacred cow's milk is enough Of what worth is even a potful of donkey's milk Even a little food given with respect is sumptuous Beloved of the Bounteous, Vema, listen!
4.	వేషభాషలెరిగి ఖాషాయవస్త్రముల్ గట్టగానె ముక్తి గలుగబోదు	Knowing external behaviour and language, wearing saffron (renunciate) clothes That does not lead to mukti (liberation)
5.	పరిగెత్తి పాలు తాగే కాంటెక్ నిల్చిని నీళ్లు తాగడం మేలు	Instead Of Running and drinking Milk, Standing and Drinking Water is Better

Performance Evaluation:

Software testing is a process of evaluating a software application or system to detect and identify defects, errors, or other issues that may impact its performance, functionality, security, or usability. According to our application we have categorized the testing process into two types Testing the model, Testing the web interface.

Testing the Model:

As we have created our model using OpenNMT, we have opted to test the model using BLUE. BLEU (Bilingual Evaluation Understudy) is a metric used to evaluate the quality of machine translation output. It measures the similarity between a candidate translation generated by a machine translation system and one or more reference translations provided by human translators.

BLEU is a widely used metric in machine translation research and development, as it provides a quick and automated way to evaluate the quality of translations generated by different systems. BLEU score ranges from 0 to 1, with 1 indicating a perfect match between the candidate and reference translations. A higher BLEU score indicates better translation quality.

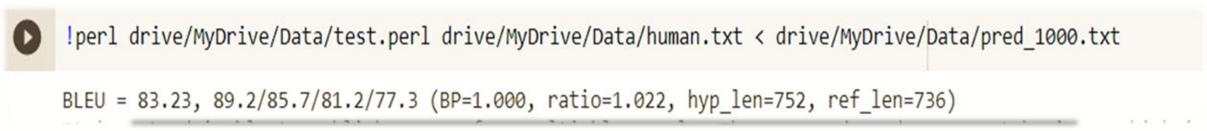
To test our model and to calculate the BLEU score we have created a file called 'test.perl' in our Google Colab, for which we have used the 'multi-bleu.perl' script provided with the Moses toolkit.

This command compares the predicted translations in test/pred.txt to the reference translations in test/tgt.txt and outputs the BLEU score, which we got as 83%.

Command:

```
!perl drive/MyDrive/Data/test.perl drive/MyDrive/Data/human.txt < drive/MyDrive/Data/pred_1000.txt
```

Output:



```
!perl drive/MyDrive/Data/test.perl drive/MyDrive/Data/human.txt < drive/MyDrive/Data/pred_1000.txt

BLEU = 83.23, 89.2/85.7/81.2/77.3 (BP=1.000, ratio=1.022, hyp_len=752, ref_len=736)
```

Fig: 4.1 BLUE score

6. COCLUSION

Summary:

India is multilingual country, where there exist more than 18 languages, there are different troops speaking different languages and people in India are not familiar with all languages they can neither understand nor speak all the languages. Most of the population in India speak languages like English and Hindi. Here we can observe a necessity of a translator whenever different people from different regions try to interact with each other, at this time we need a translator which can translate from one language to another quickly and effectively which can be achieved using a Machine Translator.

A Machine Translator is a program which can convert the text from one language to another, so the language in which the input is given is known as source language and the language in which the output is obtained is known as target language. At recent times we have different machine translators in use, which are effective and can solve this problem, but most of the times these existing translators fail when we have to convert poems, proverbs of the source language to target language. Every language has its own proverbs and poems, which are created based on the ideologies of poets, people, and culture. While translating these language specific poems and proverbs a special care must be taken, which is our problem statement, we can say our problem statement is constructing a Machine Translator which can translate different proverbs and shatakas in Telugu to English correctly. So, we can say in our case the source language is Telugu whereas target language is English. when compared to Telugu, English is morphologically simple language. So, our model translates from morphologically rich language to a morphologically simple language. To deal with this problem statement we have opted OpenNMT to build our own translation model, we have collected the data of different poems, proverbs and Telugu sentences and given this data to the Neural machine translation algorithm, Transformers. we have trained our model with the dataset and deployed in a website so that it is accessed by the user. We have designed our website using Django and SQLite.

In conclusion, building an application that can translate Telugu poems and proverbs into English is a valuable contribution to the preservation and promotion of Telugu literature. With the rise of digital technology and the increasing popularity of mobile devices, this application can provide an accessible and convenient platform for Telugu literature enthusiasts to explore the richness and beauty of Telugu literature. The application has the potential to revolutionize the way people perceive and appreciate Telugu literature, making it accessible to a broader audience across the world. Overall, the application has immense potential to not only enhance the understanding and appreciation of Telugu literature but also to promote cross-cultural exchange and understanding. The project represents an excellent example of how technology can be leveraged to preserve and promote the world's diverse cultural heritage.

Future Enhancement:

By working on this Language Translator application, I have realized there are a wide range of applications of Neural Machine Translators. We were limited to translating poems and proverbs of Telugu to English, as future enhancement I would suggest below applications

1. **Improved translation quality**: One major area for improvement would be to enhance the accuracy and fluency of the translation. By trying different approaches other than openNMT or Transformers, and building the model in different ways to compare the accuracy and improve the translation quality.

2. **Audio Translation:** Another useful feature would be to add an audio translation option, where users can listen to the translated poem or proverb in English along with its Telugu version.

3. **Customization:** Users may have different preferences for the style or tone of the translation, such as a more literal or poetic translation. So a translator could be super smart and provide the translation results according to the need of the user.

4. **Social Media Integration:** Integrating the translation application with popular social media platforms like Facebook, Twitter, or Instagram would enable users to share the translated poems and proverbs with their friends and followers.

5. **Accessible in different Telugu slangs:** It can be further upgraded towards different slangs, as people across different districts of Telangana and AP, speak differently, use different phrases and pronounce differently.

6. **Sign board Translation:** Integrating the translation application with the Artificially intelligent cameras such as PyCamera, can increase the functionality of the application, basically user can scan the board and get the translated information.

7. **Inbuilt dictionary:** For increasing the accuracy of the translations you can build a mapping dictionary of words. Which includes the information like where a particular word is used, tense of the word, synonyms, antonyms of the word etc.

8. **Extendable to different languages:** Every language has its own proverbs and poems, which are created based on the ideologies of poets, people and culture. So I believe there is an need to build a multilingual translator which is capable of understanding the proverbs, poems in every language

7. ACKNOWLEDGEMENT

We extend our heartfelt appreciation to Mrs. P. Swaroopa, our Internal Guide, for her unwavering support and for generously dedicating her time to our project. Additionally, we would like to express our gratitude to our project coordinators, Dr. K Prem Kumar, for his assistance and cooperation. We are also grateful to Dr. M.V. Vijaya Saradhi, the Head of the Department (CSE) at ACE Engineering College, for his continuous support and valuable guidance throughout our project.

8. REFERENCES

1. P.Sujatha, D. Lalitha Bhaskari " Sentence Wise Telugu to English Translation of Vemana Sathakam using LSTM
2. "OpenNMT: Open-Source Toolkit for Neural Machine Translation" by Guillaume Klein et al.
3. "Neural Machine Translation by Jointly Learning to Align and Translate" by Dzmitry Bahdanau et al.
4. "Attention Is All You Need" by Vaswani et al.
5. "Effective Approaches to Attention-based Neural Machine Translation" by Luong et al.
6. "Improving Neural Machine Translation Models with Monolingual Data" by Sennrich et al.
7. "Unsupervised Neural Machine Translation" by Lample et al.
8. "Understanding Back-Translation at Scale" by Edunov et al.
9. "Massively Multilingual Neural Machine Translation" by Johnson et al.
10. Pre-processing English-Hindi Corpus for Statistical Machine Translation (Karunesh Kumar Arora local Sharma S Agarwal centre for development of advanced computing Noida India, KIIT group of institutions, Sohna road, Bonci Gurugram India go)