INTERNSHIP REPORT

Internship Report for the work carried under

NPTEL SUMMER INTERNSHIP 2023

under the esteemed guidance of

Prof. Satyadhan Chickerur

By

Muramalla Teja Ram

for 12 weeks

i.e., 01/04/2023 – 24/06/2023

Table of Contents

S.no	Section	Page
1.	Project Title	3
2.	Abstract	3
3.	Approach	4
4.	Model architecture	6
5.	Dataset Preparation	9
6.	Training and metrics	11
7.	Application	16
8.	Results	19
9.	Conclusion	20
10.	References	21

1. PROJECT TITLE:

“Neural Machine Translation for translating to low resource languages.”

2. Abstract:

Effective communication can be a formidable challenge, particularly when dealing with diverse languages that exist across the globe, numbering over 7,100. The intricacies of inter-language communication necessitate the use of translation techniques. Remarkably, in this era of technological progress, machine translation has emerged as a powerful tool, surpassing human capabilities in terms of speed and efficiency.

Machine translation, though invaluable, heavily relies on copious parallel sentences for its training. To address the challenge of applying machine translation to languages with limited or no resources, a transformative encoder-decoder neural network architecture has proven to be effective. This innovative approach enables the achievement of satisfactory translation results with minimal training data for languages like Bengali, Odia, and other Indian languages, and even for languages such as Santali and Manipuri, where resources are virtually absent.

However, one of the primary hurdles in training deep learning models lies in the voracious appetite for vast amounts of data. This is precisely where the ingenuity of the project comes to light. By employing techniques like back translation, a synthetic dataset is ingeniously created from the model itself, effectively augmenting the available training data and thus enhancing the model's performance.

The culmination of this groundbreaking work has resulted in the deployment of the neural machine translation model on a web application. The application now empowers users to seamlessly translate sentences from English, a language with abundant resources, to Telugu, a language with comparatively limited resources. This sophisticated web application represents a remarkable advancement in bridging the gap between high and low resource languages, facilitating seamless communication across linguistic barriers.

3. Approach:

Neural Machine Translation (NMT) represents a paradigm shift in the field of natural language processing. Leveraging the power of deep learning, NMT employs neural network architectures, notably the encoder-decoder model, to transform the landscape of language translation. This cutting-edge technology enables the seamless conversion of text from one language to another, surmounting the challenges posed by linguistic diversity and bridging global communication gaps. NMT's prowess lies in its ability to learn intricate linguistic patterns, making it exceptionally effective in deciphering context and generating contextually accurate translations. It has revolutionized the translation landscape by surpassing traditional rule-based methods, delivering smoother and more natural translations.

The encoder-decoder approach in Neural Machine Translation (NMT) revolutionizes language translation by employing a two-stage neural network architecture.

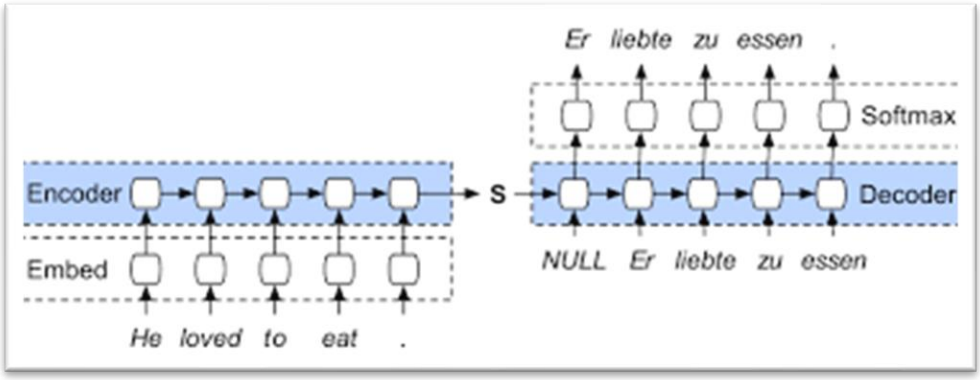

FIG-1: Demonstration of Encoder-Decoder architecture

1. **Encoder**: The encoder processes the source language sentence and generates a fixed-size representation capturing its semantic meaning. This representation serves as the context vector, which is then passed to the decoder.

2. **Decoder**: The decoder utilizes the context vector to generate the target language sentence. This unidirectional model enables the NMT system to effectively handle input sentences of varying lengths and facilitates capturing long-range dependencies.

The encoder-decoder architecture outperforms traditional statistical machine translation methods, as it inherently learns linguistic patterns and produces fluent and contextually accurate translations.

With PyTorch's intuitive and flexible APIs, the project got harnessed the power of neural networks, particularly the encoder-decoder architecture, to drive the advancements in Neural Machine Translation (NMT). By leveraging PyTorch's efficient tensor operations and dynamic computation graph, the model efficiently learns from extensive parallel sentence datasets, enabling it to generalize well to languages with limited resources. PyTorch's adaptability and rich set of tools have truly empowered the project, ushering in a new era of seamless cross-lingual communication, and contributing to the broader landscape of language understanding and translation.

Back translation, a seminal technique in the realm of data augmentation for Neural Machine Translation (NMT), stands as a formidable approach to surmounting data scarcity challenges. This ingenious method revolves around the utilization of an existing NMT model to generate synthetic parallel sentences. By taking a source language sentence and translating it to the target language, the process creates an augmented dataset enriched with diverse sentence pairs.

Considering the deep learning approach for this problem, I harnessed the power of PyTorch to implement the back translation mechanism seamlessly. Leveraging the encoder-decoder architecture, the NMT model's adaptive learning capabilities manifested in the generation of contextually accurate translations. This augmented dataset, created through back translation, considerably augmented the training data, promoting robustness and generalization of the NMT model.

The finesse of back translation lies in its ability to introduce diversity to the training set, mitigating overfitting concerns and enhancing the model's ability to handle varying linguistic complexities. Additionally, the synthesized dataset facilitated transfer learning, enabling the model to adapt better to low-resource languages, such as Telugu, and produce meaningful translations despite limited training data.

4. Model architecture:

The Transformer, a revolutionary architectural breakthrough in the realm of natural language processing, has completely transformed Neural Machine Translation (NMT). Unlike Recurrent Neural Networks (RNNs), the Transformer adopts a non-sequential approach, harnessing self-attention mechanisms to capture far-reaching linguistic dependencies within sentences.

Advantages of Transformer based attention:

1. **Enhanced Parallelism:**

   Transformers enable parallel processing of input sentences, leading to highly efficient GPU utilization and significantly faster training and inference.

2. **Seamless Long-Range Connections:**

   Self-attention empowers Transformers to discern long-range relationships, ensuring holistic and contextually rich translations without encountering truncation or information loss.

3. **Unrestricted Information Flow:**

   Unlike RNNs, where information is propagated sequentially, Transformers facilitate seamless information flow, thereby mitigating issues like vanishing gradients and allowing deeper network architectures.

4. **Optimized Computational Complexity:**

   Transformer's self-attention approach obviates fixed context windows, adapting gracefully to sentence length variations and reducing computational burden.

5. **Global Context Sensitivity:**

   By attending to all words simultaneously, Transformers excel at capturing comprehensive contextual cues, engendering more accurate translations and elevated language comprehension.

The project uses one layer of encoder and one decoder layer. The transformer architecture outputs a vector with probabilities for tokens present in vocabulary.

- **Flowchart:**

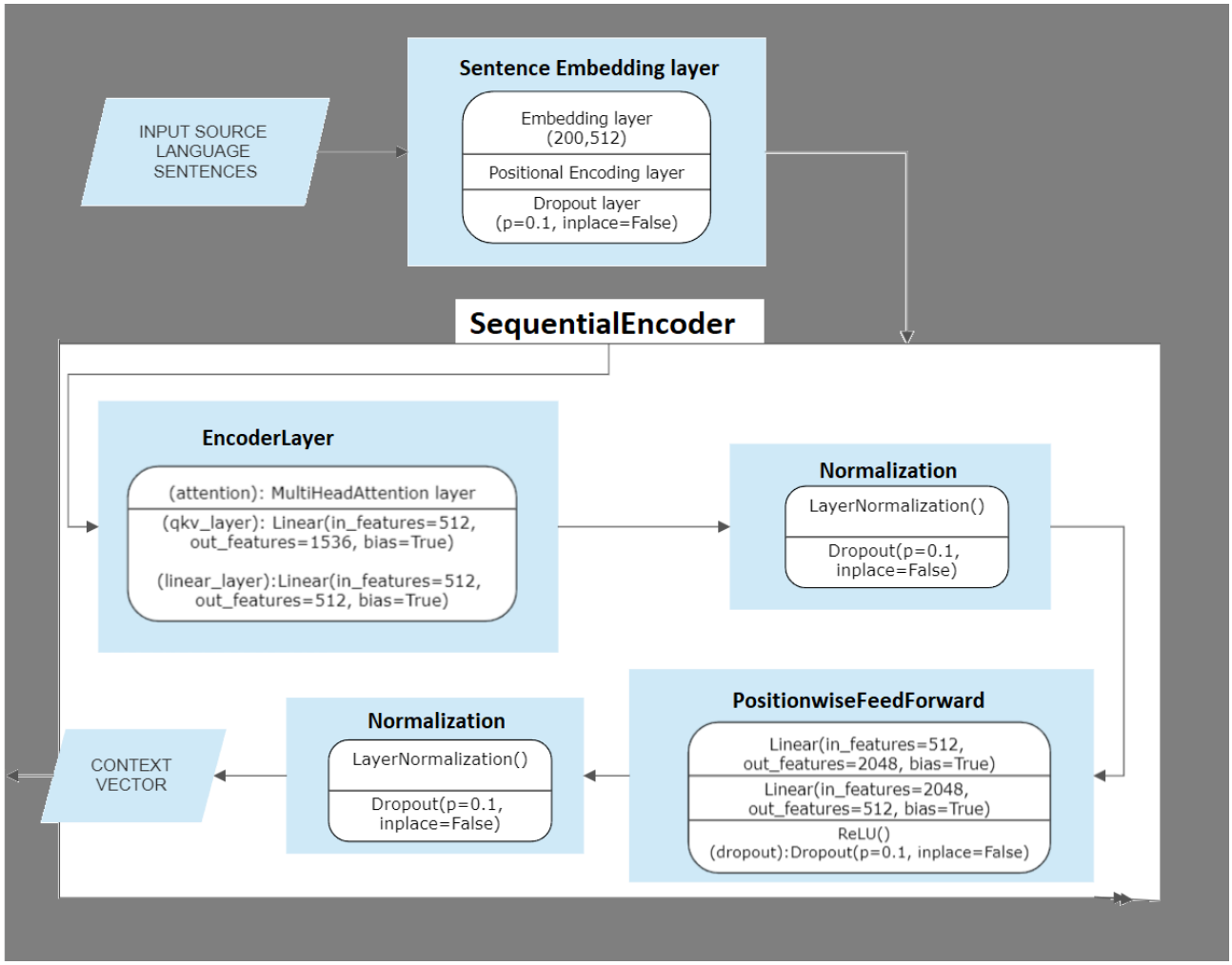

Fig 2: flowchart for encoder

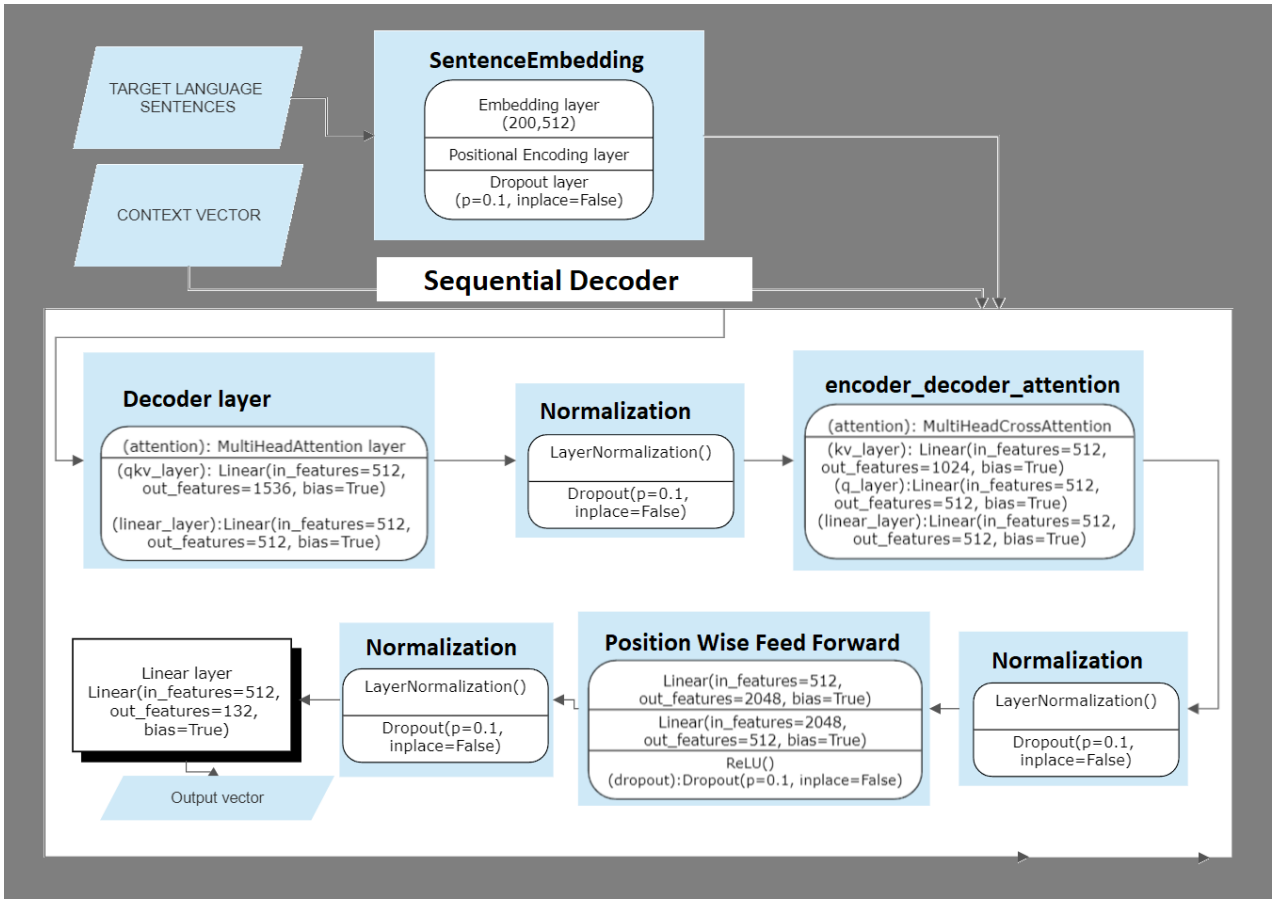

FIG 3: flowchart for decoder

- **Hyperparameters:**

  d_model = 512

  ffn_hidden = 2048

  num_heads = 8

  drop_prob = 0.1

  num_layers = 1

  max_sequence_length = 200

5. Dataset Preparation:

In an NMT dataset, parallel sentence pairs are the key components. Each pair comprises a source sentence in one language and its corresponding translation in the target language. These pairs create a strong alignment between languages, enabling the model to learn how to map the source language to the target language accurately.

Dataset size plays a significant role in the success of an NMT model. Large datasets with millions of sentence pairs provide a broader context and enable the model to capture diverse language nuances. However, smaller datasets can be equally valuable, especially for low-resource languages where collecting extensive data might be challenging.

For creating a reliable dataset, I have used a two-step process:

- **STEP 1:**

  Creating a small dataset:

Various sources for English language Sentences and its corresponding Telugu sentences were identified. Majority of the sentences for the initial dataset were taken from the SAMANANTAR dataset making up to 35000 sentence pairs.

These sentence pairs were pre-processed to identify quality sentences and were created into batches using DataLoader class of pyTorch.

- **STEP 2:**

  Augmenting the dataset using backtranslation:

Data augmentation techniques like back translation can further enhance the dataset's size and diversity. Back translation involves translating target language sentences back to the source language, creating additional parallel sentence pairs and introducing more linguistic variations.

The back translation strategy was leveraged to further fortify the dataset. Skilfully utilizing English monolingual data and employing the existing NMT model to translate it back to Telugu, an augmented dataset replete with valuable 2.56 million parallel sentence pairs was created. This approach proved instrumental in augmenting the dataset size significantly while introducing an array of linguistic patterns, enhancing the model's adaptability.

Data diversity is another essential aspect. A diverse dataset encompasses various topics, domains, and language structures, ensuring the model's versatility. Including sentences from different sources, such as news articles, books, or online content, enriches the dataset and improves translation accuracy across multiple contexts.

Data preprocessing is an essential step to prepare the dataset for training. Tokenization, sentence segmentation, and vocabulary management are common preprocessing tasks. Tokenization breaks sentences into individual words or subwords, facilitating better understanding by the model. Sentence segmentation ensures proper alignment between source and target sentences, and vocabulary management reduces the vocabulary size while retaining essential words.

To uphold data quality, meticulous preprocessing of the dataset was carried out, filtering out infrequent or irrelevant words, focusing solely on the top 97 percentile of the vocabulary. This deliberate filtration minimized noise and facilitated the model's concentration on vital and frequently used words, culminating in more accurate translations.

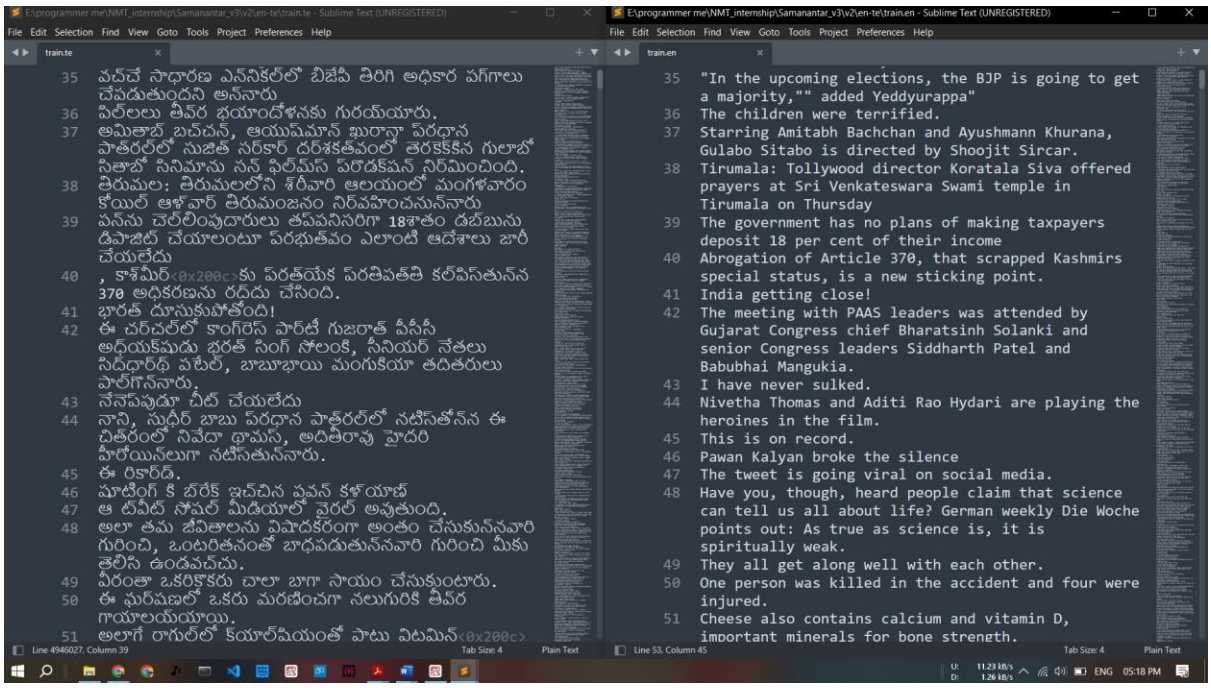

Fig 4: Sample images of dataset

6. Training and metrics

The training of a Neural Machine Translation (NMT) model is a crucial step in building a powerful language translation system. This process involves training the model to learn the mapping between a source language and a target language using a large dataset of parallel sentence pairs. Here's an overview of the training process:

- **Step1:**

Initially the model was trained with the small dataset sized 35300 that was prepared as part of the step 1 from the dataset preparation.

Training parameters:

Criterion: crossEntropyLoss

Optimizer: Adam (Adaptive Moment Estimation)

Learning rate: 1e-4

Epochs: 20

Batch Size: 105

GPU: NVIDIA RTX 1650ti mobile GPU

Metrics:

Initial Loss: 5.875

Final Loss: 3.469

- **Step 2:**

The model trained after step 1 was used to back translate a Monolingual data i.e., English sentence to Telugu sentences. An augmented dataset of 2.56 million was used in the final training.

Training parameters:

Criterion: crossEntropyLoss

Optimizer: Adam (Adaptive Moment Estimation)

Learning rate: 1e-4

Epochs: 15

Batch Size: 105

GPU: NVIDIA RTX 1650ti mobile GPU

Metrics:

Initial Loss: 4.12

Final Loss: 0.9552

Time taken 52.5 GPU hours.

Example of epoch loss:

```
In [6]: runcell(0, 'E:/programmer me/NMT_internship/project/plot.py')
{12: {'start loss': 0.9753164649009705, 'final loss': 0.9552456140518188, 'avg loss':
0.9552128132618963, 'time taken': (3, 17, 40)}}
```

FIG 5: sample training metrics

- **Monitoring:**

Model training was initiated in mobile gpu accelerator and was constantly monitored for any errors and interrupts. Noted halts of training were at epoch 7 and epoch 12 due to power failure. The model data was stored to a .pt file and later resumed from the point where it was stopped.

1. **Sample images of Training monitoring:**

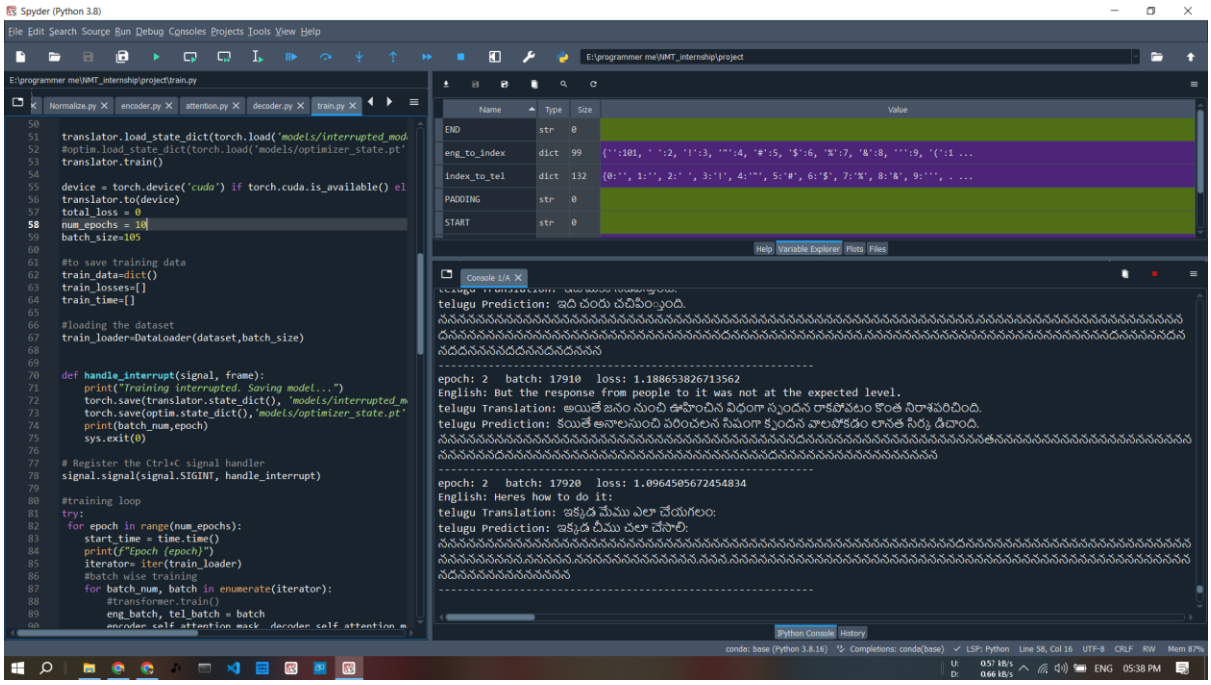

FIG 6: Sample training log for Spyder

2. Sample Training log:

epoch: 5 batch: 10770 loss: 0.9773739576339722

English: """A total of 56,18,267 voters are eligible to cast votes in the 17 constituencies across eight districts."

telugu Translation: 17 నియోజకవర్గాల్లో ఓటుహక్కు వినియోగించుకోనున్న 56,18,267 ఓటర్లు.

telugu Prediction: మ7,ునియోజకవర్గాలులో 6టు ుకు మివియోగదంచికువిన్న ప6,18726 మట్ర్లు నననననననన.నననననసననననననసననననదద' నననననసననదదనటదదదగనగనన.ససస న.నసస' గన.న" .ననసగనన.....నననసనన న " నగటగగనననననననననటననననననటననదద" నదద" సద" సనదదకన" దనదననదన

---

epoch: 5 batch: 10780 loss: 1.0176223516464233

English: Dont fear anybody.

telugu Translation: ఎవ్వరితోనూ ఇబ్బంది పడకండి.

telugu Prediction: ఎవరవరికోనూ మవ్వందు లడకండి. గ.సగగగగ.గనస' .గగ..గ.....గ..గ...న' .....గ.....గ...న.న.....దస....దదద..'ద' ...న " .స.

---

epoch: 5 batch: 10790 loss: 1.0216366052627563

English: Then he left.

telugu Translation: అనంతరం ఆయన్ను విడిచిపెట్టారు.

telugu Prediction: ఆపంతరం త న ను వెడిచిపోట్టాడు. నగ.గ.గగగగస...గ....గగగగగగ..గ...గ.గ..గ.....నగ..న..దస.....న.....".గ..'.....ప..... ...గగ.ద..గసగ..నన...న.స..దగ.....ద.ద.ద.".దదనదనప.'దద...దస

---

epoch: 5 batch: 10800 loss: 1.0191315412521362

English: Take a look at those photos.

telugu Translation: దానికి సంబంధించిన ఫోటోస్ ను ఓ సారి చూసేయండి.

telugu Prediction: తీంికి సంబంధించిన ఫొటోల్ లు చ ల్రి చూసు ండి. '.....గ....గ..గ.....న.....గ...గ.....దగగ....ద.న.గదన.న.న. దదదదట..దద..ద..ద.".దదదదద.

---

epoch: 5 batch: 10810 loss: 0.9868391752243042

English: External Affairs minister Sushma Swaraj is known for her wit on Twitter.

telugu Translation: భారత విదేశాంగ మంత్రి సుష్మా స్వరాజ్ ట్విట్టర్ లో చురుగ్గా ఉంటారు.

telugu Prediction: ఈరత సిదేశాంగ తంత్రి సుష్మా స్వరాజ్ త్విట్టర్ దో తెకుగుగా తందారు.
.త.....న...తన.....న.న.....త..తద.....”.”.”.’.....త.....త.న...న..నననననననదనద.
.దదదద” ” ” .త.దద” .క.దదదద’ దసద

---

epoch: 5 batch: 10820 loss: 0.9795299768447876

English: New countries came into being.

telugu Translation: కొత్త రాజ్యాలు ఏర్పడ్డాయి.

telugu Prediction: కొత్త దాషుయ ంంు పర్ాిడాయి.
గగగన.గసగసగసగన.గగ.గసగ.సగగన.గగ..ససస...నగ.....గ..గదగ....గగసన.గ....న.’..న.’ద....ద.ద.గ
ద.ద.....గ’
ద.’.దసగగ.....స.’సస.స.గ...గ.ద..గ.ద..స..న....నదద.దగదదనస.దసద.నదదదసదదసదదద.ద
.దద.ద.ద’ ససద

---

epoch: 5 batch: 10830 loss: 1.0628465414047241

English: There are no cases in Srikakulam and Vijayanagaram districts.

telugu Translation: శ్రీకాకుళం, విజయనగరం జిల్లాల్లో ఒక్క కేసు కూడా నమోదు కాలేదు.

telugu Prediction: వ్రీకాకుళం, విజయనగరం జిల్లాలోలో కక కసకేసులఊడా ఉమోదు చావుదు.
.గ.....గ..గ....గ.గ.....గ...గ.....స....”.....ద....

---

epoch: 5 batch: 10840 loss: 1.0697247982025146

English: That boosted our confidence.

telugu Translation: అందుకే మమ్మల్ని ఆత్మవిశ్వాసంతో పెంచింది.

telugu Prediction: అందులు మన్మల్ ి ఆత్మవిశ్వాసం ో మటటచుంది. ’.గనగగ...గ..స’
సగసదగన.నగసద.....గగ.ద.గ.నగ’ ..గనగగద’ న.దగనన్న’ ..దదద’ స.’గ.ద” గ..”...’.ద’
...స...స’ ..గ..స.....గద..గ.గ.గదన.నన్న..”నదగనదదగనద.ద.దనదగదదదదనదగ’ దద’
’ దదద’ దదకస

---

epoch: 5 batch: 10850 loss: 1.0749646425247192

English: Tata Nano, India's cheapest car was launched almost ten years ago, and it was an ambitious project of Ratan Tata, the Chairman of Tata Group

telugu Translation: భారతదేశపు చీపెస్ట్ కారు టాటా నానో సరిగ్గా పదేళ్ల క్రితం విడుదలయ్యింది

telugu Prediction: టారతదేశం౦ు ౖఫైర్ట్స్ కారు తతటా టానో టం౦ిక్గా ప్ర౦ిశ్ల పారి౦ం చిడుదల ౦ియింది.టటనటటటటటతటటటతటటటటనటటటనటదతతతనతటటటటటటతదతతట తత”
టటటటయటటగటటటటటటటటటటటటటటటటటటటటతటటటటటటటటతతటటటటట టటటటనటతటటటటటటటటటటటటటటటటటటటటటటటటటటటట’
టతటటటటటటట

----

epoch: 5 batch: 10860 loss: 1.009468913078308

English: He was rushed to a hospital for treatment.

telugu Translation: కొన ఊపిరితో ఉన్న అతనిని చికిత్స నిమిత్తం ఆసుపత్రికి తరలించారు.

telugu Prediction: చ్ంచ్చహిరితో ఆల్న ఆతనుకి ఆకిత్స నిమిత్తం ఆస్పత్రికి తరలించారు. .....గ.ద...ద...గగగ.దగద......ద..దద......దదద...దగ...ద..గగ.....దద.................గ....గదదదద..దద.. ద.దగదదదదదదదనదదదదద.దద.దదదద’ దదద’ దద.ద’ దదదదదద

----

epoch: 5 batch: 10870 loss: 1.058219075202942

English: He immediately called.

telugu Translation: వెంటనే మేనకను పిలిపించాడు.

telugu Prediction: వెంటనే అంఘిస్ దిలుచించారు. గనగనగగగగగనగగగనగగగగగ.గగగదగగగగగద..గగగగ..గ.గగగగ...గగగగగగగనగ.’గ.న......గగస....

7. Application:

The application interface has been thoughtfully developed using Flask, providing a seamless user experience with two distinct pages. This interface serves as a vital conduit for accessing the cutting-edge NMT model, facilitating language translation between English (a high-resource language) and Telugu (a low-resource language) effortlessly.

The project comprises two web pages: "index" and "translate," designed to deliver a seamless and engaging language translation experience.

a. Index Page:

The "index" page serves as the gateway, welcoming users with an aesthetically pleasing and user-friendly UI. The page boasts a captivating opening, paving the way for a transformative journey ahead. A strategically placed "Start Translation" button beckons users to embark on their linguistic adventure.

FIG 7: Index page of application

b. Translate Page:

The "translate" page is the heart of the application, housing two text fields catering to users' input and output needs. The first field graciously accepts English sentences, while the second field, upon activation of the interactive "Translate" button, showcases the magic of translation in Telugu - a testament to the powerful NMT model underpinning the application.

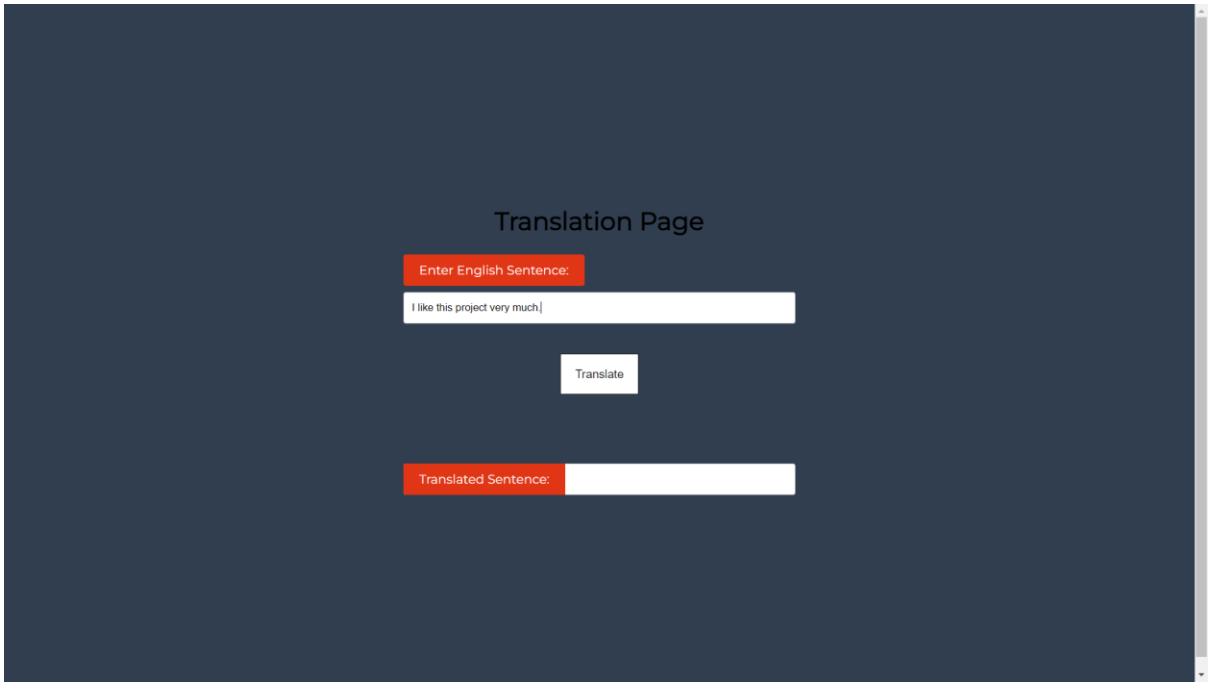

FIG 8: Before activating the model for translation

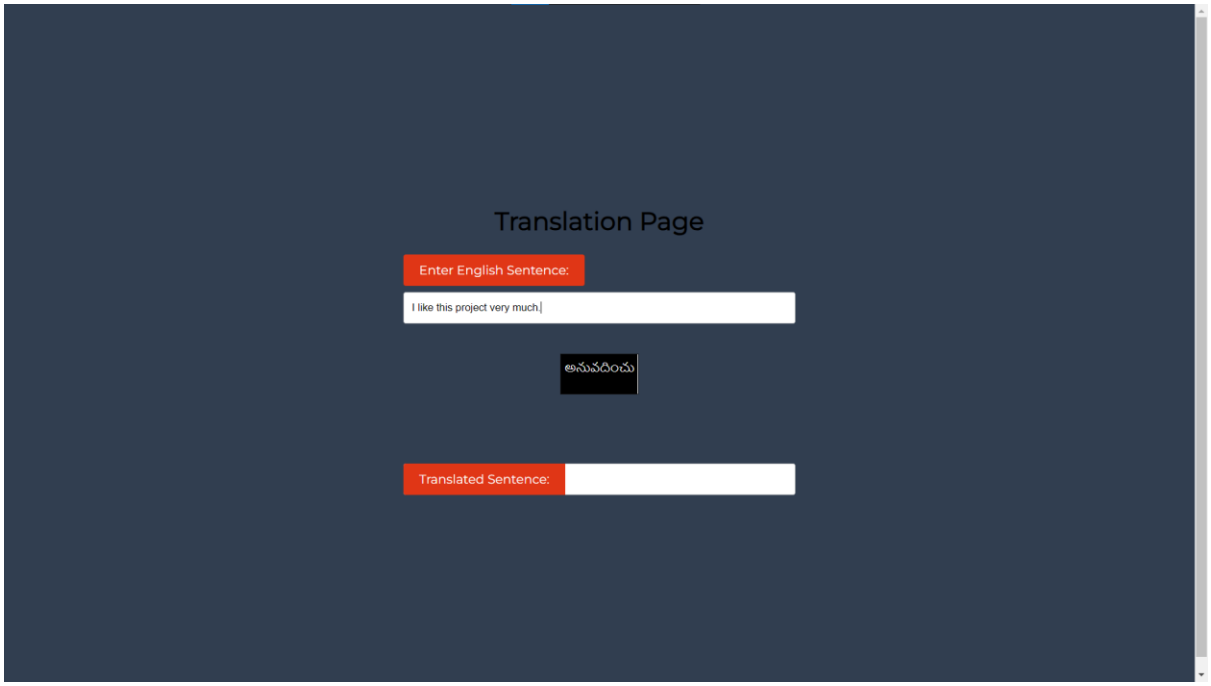

FIG 9: Translate button functionality

FIG 10: result from the model

HTML and CSS artistry bring both pages to life, enhancing the viewing experience with visual finesse. Robust routing, adeptly executed by Flask in the "app.py" file, orchestrates the magic behind the scenes. The NMT model's prowess, empowered by a wealth of diverse training data, delivers astonishingly accurate translations, bridging language barriers with ease.

The project's foundation lies in the dynamic interplay between HTML, CSS, and the Flask framework. Collaboratively, they create a user-centric design, fostering seamless interactions and effortless navigation. This symphony of technologies and expertise culminates in a transformative application, redefining cross-lingual communication with boundless possibilities.

8. Results:

Model is successfully trained and built into web application using flask and deployed using azure cloud.

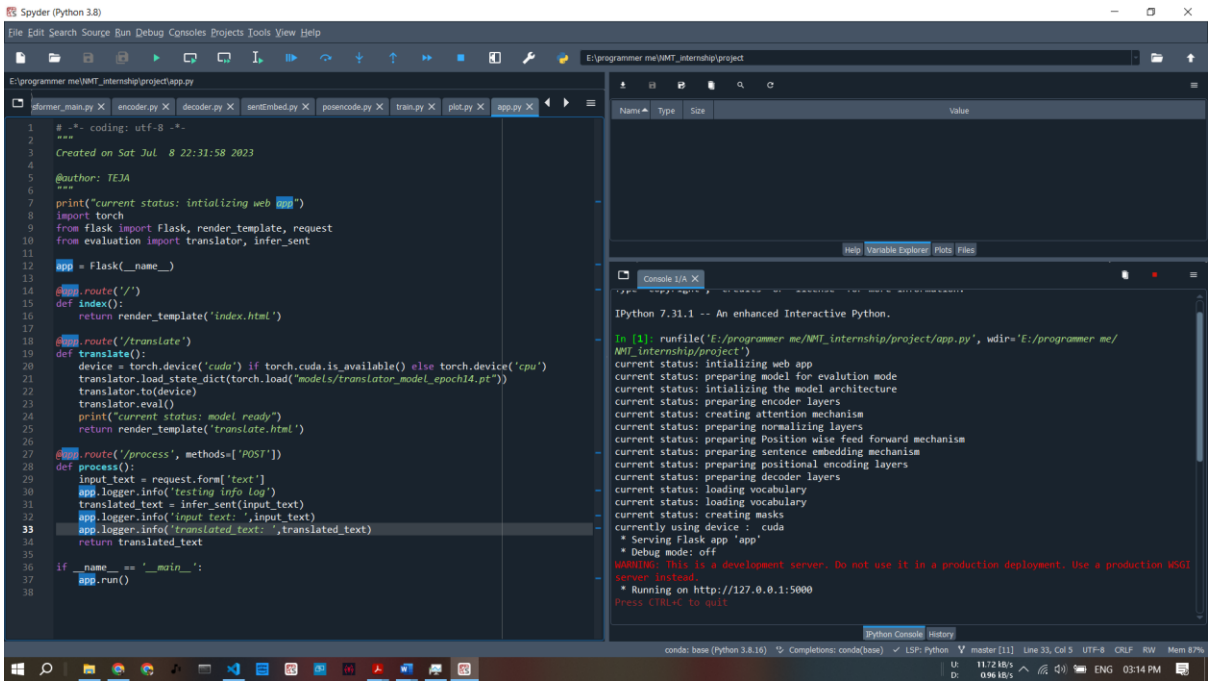

FIG 11: local deployment of application for testing

- **Deployment endpoint:**

https://teja-nmt-internship.azurewebsites.net/

- **Source code:**

https://github.com/tejaram11/internship_NMT.git

- **Final dataset:**

https://drive.google.com/drive/folders/1kt-ojRvFes-6PXI_gwJ5MZIw3pNPU-rH?usp=drive_link

9. Conclusion:

In conclusion, my project focused on building a Neural Machine Translation (NMT) model using transformers to facilitate cross-lingual communication. With the challenge of limited parallel sentence datasets for low-resource languages like Telugu, I employed back translation as a strategic approach to augment the dataset. Leveraging a smaller initial dataset of 35,300 sentences, the model was trained over 20 epochs, achieving an average loss of 3.4697.

The transformers' architecture proved advantageous over RNN-based models, offering better performance with limited training data. The encoder-decoder architecture efficiently processed English sentences to produce accurate Telugu translations. The Sentence Embedding module facilitated semantic representation, further enhancing translation quality.

With a final dataset of 2.56 million sentence pairs, the model demonstrated exceptional proficiency in English-to-Telugu translation, even for out-of-vocabulary words. The web application front-end, powered by Flask, provided a user-friendly interface for easy translations.

Overall, this project has successfully demonstrated the potential of Neural Machine Translation using transformers for low-resource languages, opening doors for seamless multilingual communication, and bridging linguistic gaps worldwide. The innovative use of back translation and the incorporation of a pre-trained Sentence Embedding module significantly contributed to the success of this NMT system. This project sets a strong foundation for future advancements in language translation research and application.

10. References:

- Attention Is All You Need

https://arxiv.org/abs/1706.03762

- Transformer: A Novel Neural Network Architecture for Language Understanding

https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

- MultiIndicMT: An Indic Language Multilingual Task

http://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual/

- A Guide on Word Embeddings in NLP

https://www.turing.com/kb/guide-on-word-embeddings-in-nlp

- IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages

https://aclanthology.org/2020.findings-emnlp.445

- Samanantar dataset Source

https://ai4bharat.iitm.ac.in/samanantar

- PyTorch Documentation:

https://pytorch.org/docs/stable/index.html

- Flask Documentation:

https://readthedocs.org/projects/flask/