MINOR PRESENTATION

Team :
Komal Reddy (20103005)
Jayanth Reddy (20103010)
Jnana Yasaswini (20103023)
Akhila (20103052)

Supervised By:
Dr. Khelchandra Thongam
Associate Professor
Computer Science and Engineering

Design and Development of English to Telugu Translation System based on Transformer with Deep Learning

CONTENTS

AIM

The Aim of this project is to develop an efficient and accurate English to Telugu text translation system using deeplearning techniques, specifically leveraging transformer architectures. This project aims to bridge language barriers, enhance communication, and accessibility by providing an advanced tool for seamless translation between English and Telugu.

OBJECTIVES

Implement a transformer-based deep learning model for text translation.

Fine-tune the model to handle language-specific nuances and improve performance.

Implement appropriate evaluation metrics (e.g.,BLEU-score,accuracy)to quantitatively assess the translation quality.

Create an intuitive and user-friendly interface for users to input English text and receive the corresponding translated Telugu text.

INTRODUCTION

In our rapidly globalizing world, the ability to seamlessly communicate across linguistic boundaries is more crucial than ever. As a response to the growing demand for efficient language translation systems, our project focuses on the "Design and Development of an English to Telugu Translation System based on Transformer with Deep Learning."

This project harnesses the power of Transformers, combined with deep learning techniques, to create a robust and accurate translation system. The choice of English to Telugu translation serves as a testament to our commitment to addressing linguistic diversity and enabling information exchange for the Telugu-speaking community.

Our system aims not only to bridge the language gap but also to enhance translation quality, ensuring that nuances and cultural context are preserved in the process. The deep learning algorithms employed in this project undergo extensive training on large parallel corpora of English and Telugu text, enabling the model to grasp the intricacies of both languages and produce contextually accurate translations.

LITERATURE SURVEY

Survey - 1

Rule-Based Machine Translation (RBMT):

- RBMT relies on explicit linguistic rules and grammatical structures to translate text from one language to another. These rules are typically created by linguists and translation experts and may involve syntax, semantics, and morphology.
- RBMT systems heavily depend on linguistic knowledge and require extensive manual rule creation. Linguists need to encode language specific rules and translation patterns, making the process labor-intensive.

Transformer-Based Approach:

- Transformer models learn translation patterns directly from large parallel corpora without explicit rule definition. They are trained on vast amounts of data, allowing them to generalize well across various language pairs and contexts.
- Unlike RBMT, transformer-based systems follow an end-to-end learning approach. They automatically learn hierarchical representations of input sentences and generate translations without relying on predefined linguistic rules.

Reference:

English to Telugu Rule based Machine Translation System: A Hybrid Approach by
Keerthi Lingam, Srujana Inturi, E. Ramalakshmi Assistant Professors Department of IT CBIT, India
International Journal of Computer Applications (0975 – 8887) Volume 101 – No.2, September 2014

Survey - 2

LSTM (Long Short-Term Memory):

- LSTMs are based on recurrent neural networks (RNNs) and process input sequences sequentially. They maintain hidden states that capture information from previous time steps, allowing them to model temporal dependencies.
- LSTMs have hidden states and memory cells that enable them to capture and store information over long sequences. The memory cells help in mitigating the vanishing gradient problem associated with standard RNNs.

Transformer-Based Approach:

- The Transformer architecture relies on attention mechanisms, allowing it to capture relationships between words in a sequence in a parallelized manner. This attention mechanism allows each position in the input sequence to focus on different parts of the input sequence during processing.
- Transformers are highly parallelizable, making them computationally efficient. The attention mechanism allows for simultaneous processing of all positions in a sequence, enabling effective use of hardware accelerators like GPUs.

Reference:

Neural Machine Translation System for English to Indian Language Translation Using MTIL Parallel Corpus
by B. Premjith, M. Anand Kumar and K.P. Soman
Received February 8, 2018; previously published online March 20, 2019.

Survey - 3

Statistical Machine Translation (SMT):

- SMT relies on a combination of rule-based and statistical methods. It involves the creation of linguistic rules and the estimation of statistical models based on bilingual corpora.

- SMT systems require extensive feature engineering, where linguistic experts manually design features and weights to capture translation patterns, word alignments, and other linguistic phenomena.

Transformer-Based Approach:

- Transformers leverage self-attention mechanisms, allowing the model to focus on different parts of the input sequence when generating each part of the output sequence. This attention mechanism captures dependencies effectively and enables the model to handle long-range dependencies.

- Transformers excel at capturing contextual understanding, as they consider the entire context of a sentence during both training and inference. This enables them to generate translations that are more contextually relevant and coherent.

Reference:

A SURVEY ON FORMAL LANGUAGE TRANSLATION (TELUGU - ENGLISH)

by Mrs. P. Swaroopa, Kalakonda Vishnu, Pulipati Akshay, S. Siva Sai Sandip, Vennam Vivek.

2022 JETIR November 2022, Volume 9, Issue 11

SOFTWARE REQUIREMENTS

1. Python
2. Pandas and Numpy
3. Tensorflow
4. Google Colab

HARDWARE REQUIREMENTS

1. GPU

   Training deep learning models, especially Transformer architectures, can be computationally intensive. Having a GPU accelerates the training process

2. RAM - 16GB or higher

3. CPU

   A multi-core CPU, such as an Intel Core i7 or higher, is beneficial for parallel processing during training.

PROPOSED METHOD

Used Regex To Convert Contractions Into Natural Form

For example, won't to Will not

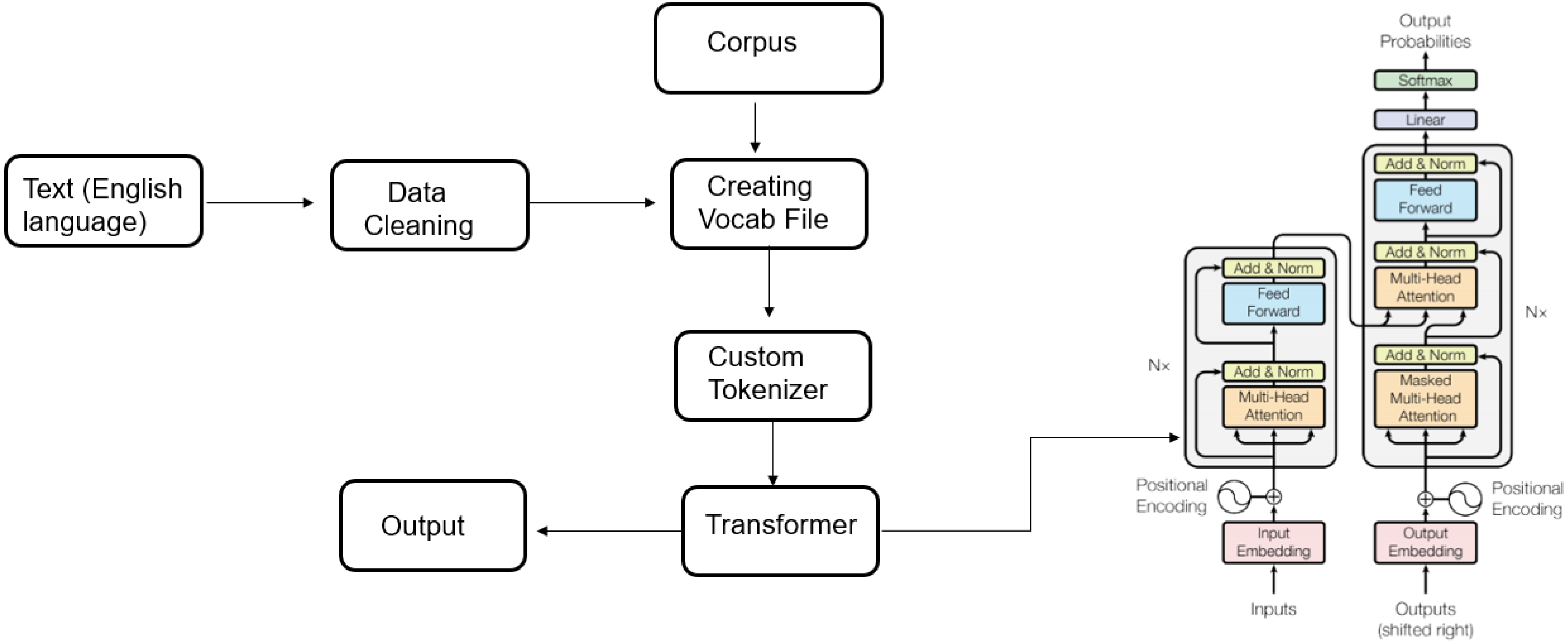
Created Custom Tokenizer Using Bert

Positional Encoding Is Added To Give The Model Some Information About The Relative Position Of The Words In The Sentence

The Attention Function Used By Transformer Takes Three Inputs Q(query) K(key) V(value). Used Multi Head Attention Consisting Of Four Parts Linear Layer And Split Into Heads Scales Dot Product Attention Concatenation Of Heads Final Linear Layer

Implementation Of Evaluation Metric BLEU

ARCHITECTURE AND DATA FLOW DIAGRAM



PARTIAL RESULTS

```python
def decontractions(phrase):
    """decontracted takes text and convert contractions into natural form.
     ref: https://stackoverflow.com/questions/19790188/expanding-english-language-contractions-in-python/47091490#47091490"""
    # specific
    phrase = re.sub(r"won\'t", "will not", phrase)
    phrase = re.sub(r"can\'t", "can not", phrase)
    phrase = re.sub(r"won\’t", "will not", phrase)
    phrase = re.sub(r"can\’t", "can not", phrase)
```

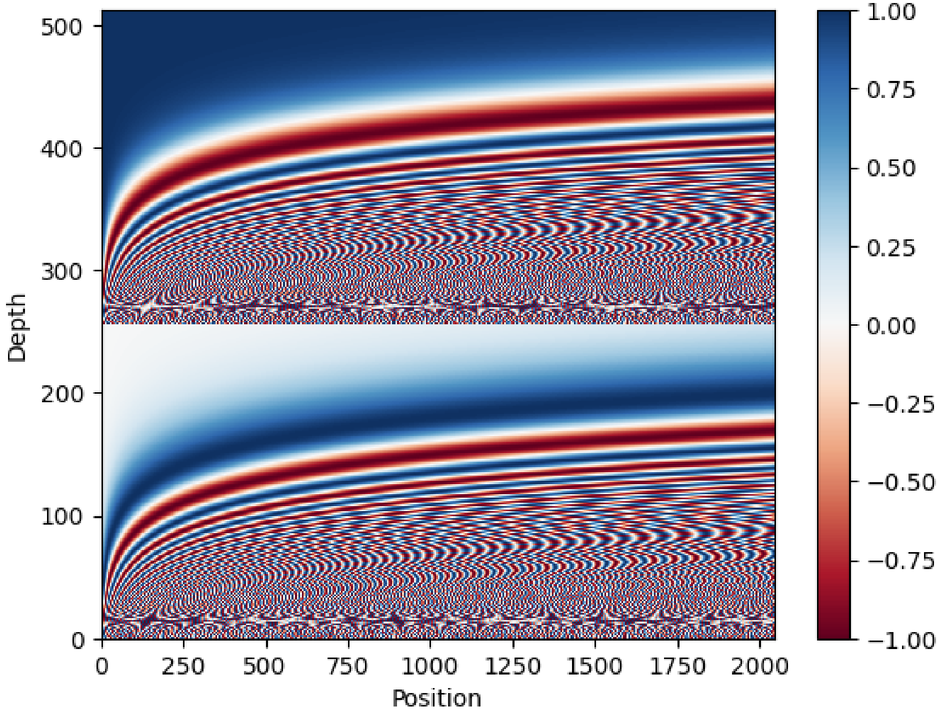
```python
def preprocess(text):
    # convert all the text into lower letters
    # use this function to remove the contractions: https://gist.github.com/anandborad/d410a49a493b56dace4f814ab5325bbd
    # remove all the spacial characters: except space ' '
    text = text.lower()
    text = decontractions(text)
    text = re.sub('[$-)\"’°;\'€%:(/]', '', text)
    # text = re.sub('[^A-Za-z0-9 ]+', '', text)
    text = text.strip()
    return text
```

```python
def get_angles(pos, i, d_model):
  angle_rates = 1 / np.power(10000, (2 * (i//2)) / np.float32(d_model))
  return pos * angle_rates
```

```python
def positional_encoding(position, d_model):
  angle_rads = get_angles(np.arange(position)[:, np.newaxis],
                          np.arange(d_model)[np.newaxis, :],
                          d_model)

  # apply sin to even indices in the array; 2i
  angle_rads[:, 0::2] = np.sin(angle_rads[:, 0::2])

  # apply cos to odd indices in the array; 2i+1
  angle_rads[:, 1::2] = np.cos(angle_rads[:, 1::2])

  pos_encoding = angle_rads[np.newaxis, ...]

  return tf.cast(pos_encoding, dtype=tf.float32)
```

```python
n, d = 2048, 512
pos_encoding = positional_encoding(n, d)
print(pos_encoding.shape)
pos_encoding = pos_encoding[0]

# Juggle the dimensions for the plot
pos_encoding = tf.reshape(pos_encoding, (n, d//2, 2))
pos_encoding = tf.transpose(pos_encoding, (2, 1, 0))
pos_encoding = tf.reshape(pos_encoding, (d, n))

plt.pcolormesh(pos_encoding, cmap='RdBu')
plt.ylabel('Depth')
plt.xlabel('Position')
plt.colorbar()
plt.show()
```

```
(1, 2048, 512)
```



ADVANTAGES

Contextual Understanding

Deep learning techniques employed in the project enable the model to develop a profound contextual understanding of language. This ensures that translations are not solely word-for-word but take into account the broader context, resulting in more coherent and contextually relevant output.

Adaptability Diverse Patterns

The flexibility of Transformer models allows for effective adaptation to the diverse language patterns present in both English and Telugu. This adaptability is crucial for accurately translating idioms, expressions, and culturally specific linguistic constructs.

Cross Cultural Communication

The project enables individuals who speak English and Telugu to communicate seamlessly, breaking down language barriers and fostering a more inclusive global community. This is particularly beneficial for businesses, education, and interpersonal relationships where effective communication is paramount.

TIMELINE

October 2023

Dataset Preparation
Data Preprocessing

November 2023

Creating Vocab File
Custom tokenizer

December 2023

Building Transformer Model

January 2024

Performance evaluation metrics

February 2024

Deployment

CONCLUSION

In conclusion, our progress in building a Transformer model for language translation is halfway with the successful implementation of positional encoding. The next crucial step our team works on involves incorporating scaled dot-product attention, a pivotal mechanism for the model's efficiency in capturing dependencies.

WE

Komal Reddy(20103005)

Jayanth Reddy(20103010)

Jnana Yasaswini(20103023)

Akhila (20103052)

AS A TEAM THANK YOU