greatlearning
Learning for Life

March, 9 - 2025

Capstone April 2024 Batch Group - 23

Interim Report

NLP Machine Translation Project

Prepared by:
Bhagawan Das Armani, Gunasekaran Venkatesan,
Madhuman Basu, Sanjay R, Vinayak Hampiholi, Vipin Nair

1. Summary of Problem Statement, Data, and Findings

Problem Statement

The objective of this project is to develop a Machine Translation Model that translates between English and German. The dataset is derived from the ACL2014 Ninth Workshop on Statistical Machine Translation, which contains parallel text in German-English from three sources:

- Europarl v7 (Formal parliamentary proceedings)
- Common Crawl (Informal, web-crawled text)
- News Commentary (News and journalistic content)

Findings

- The dataset contains a mix of formal and informal text, making it diverse and challenging for a translation model.
- The data is imbalanced, with varying sentence lengths and vocabulary sizes across different sources.
- Initial observations showed duplicate sentences, missing values, and non-German/English sentences that needed cleaning.
- Simple RNN and LSTM models were designed for translation, with BLEU scores computed for performance evaluation.

2. Summary of the Approach to EDA and Pre-processing

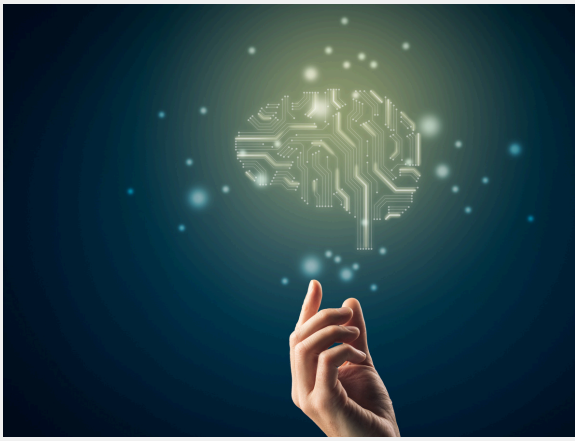
Exploratory Data Analysis (EDA) & Visualisations

- Word Count Distributions: Showed sentence length variations across datasets.
- Word Clouds: Highlighted the most frequently used words in English and German.
- Data Statistics: Summarized the number of sentences, unique words, and outliers.

Pre-processing Steps Taken

1. Data Cleaning:
    - Removed missing values and duplicate sentences.
    - Filtered out non-German/English text.
2. Tokenization & Normalization:
    - Used spaCy for faster tokenization and lemmatization.
    - Removed stopwords and special characters.
3. Dataset Reduction for Faster Execution:
    - Limited training dataset to 5,000 samples to improve execution time.
    - Capped vocabulary size at 5,000 words.
    - Reduced max sentence length to 20 words to optimize memory usage.

3. Deciding Models and Model Building



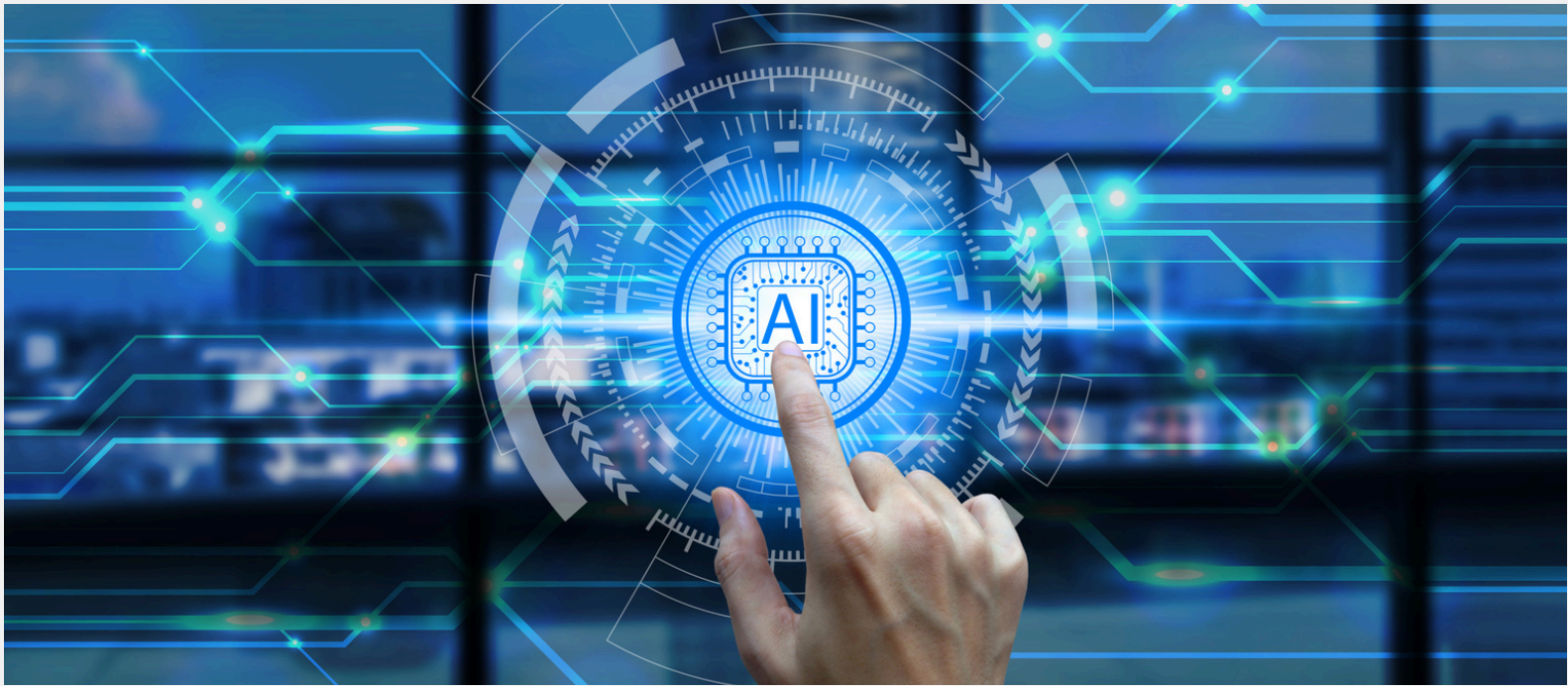
Model Choices
- Simple RNN Model (Baseline Model)
- LSTM Model (Better for sequential learning)

Training Process
- Embedding Layer: Converts words into vector representations.
- RNN/LSTM Layer: Processes sentence sequences.
- Dense Output Layer: Predicts translated words.

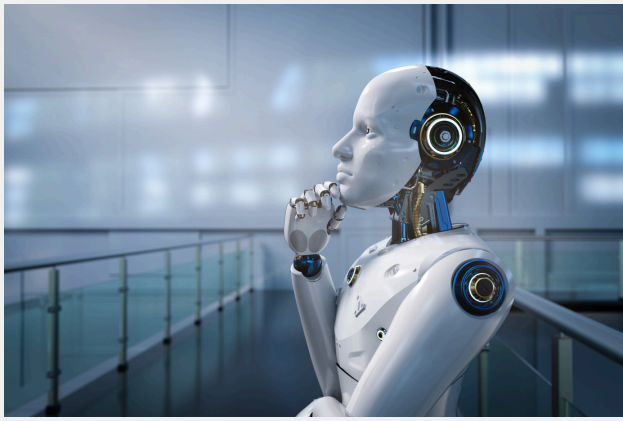
Model Performance
- Models trained on a small dataset to reduce execution time.
- Evaluation Metric: BLEU Score to measure translation accuracy.

Results:
- RNN BLEU Score: 0.092
- LSTM BLEU Score: 0.0

RNN performed better than LSTM, but BLEU scores remain low due to limited training data and short training duration.



4. How to Improve Model Performance?



Possible Enhancements

Increase Training Data → Expand dataset to 10,000+ sentences.
Train for More Epochs → Increase from 3 epochs to 10+.
Use Pretrained Word Embeddings → Such as FastText or Word2Vec.
Implement an Attention Mechanism → Helps focus on key words in long sentences.
Hyperparameter Tuning → Adjust learning rate, dropout, and batch size.

Next Steps

Improve the model performance by increasing training time and dataset size.
Explore Transformer-based models (e.g., BERT, T5) for better translation.
Deploy the best-performing model for real-world testing.

Code Implementation & Visualisations

The project implementation includes:

- Code for data cleaning, NLP preprocessing, and model training.
- Visualisations for data insights (word clouds, sentence length distribution).
- Performance evaluation using BLEU scores.
- Model saving for future inference.

This report summarises Milestone-1 and sets the direction for further improvements in machine translation modeling.

PROJECT TASK: MILESTONE 1

Step 1: Import and merge the three datasets.

Load and Verify Datasets

```
[22] import pandas as pd
     import matplotlib.pyplot as plt
     from wordcloud import WordCloud
     from IPython.display import display, Markdown

     # Function to load parallel datasets with error handling
     def load_parallel_data(filepath_de, filepath_en, dataset_name):
         """Loads English-German parallel datasets and handles line mismatches."""

         try:
             # Read German text
             with open(filepath_de, "r", encoding="utf-8") as de_file:
                 de_lines = de_file.readlines()

             # Read English text
             with open(filepath_en, "r", encoding="utf-8") as en_file:
                 en_lines = en_file.readlines()

             # Find the minimum number of lines
             min_length = min(len(de_lines), len(en_lines))

             # Truncate the longer file to match the shorter one
             de_lines = de_lines[:min_length]
             en_lines = en_lines[:min_length]

             # Create DataFrame
             df = pd.DataFrame({"German": de_lines, "English": en_lines})

             # Display confirmation
             display(Markdown(f"### Successfully Loaded {dataset_name} Dataset"))
             display(Markdown(f"Total Sentence Pairs (After Fixing Mismatch): {len(df)}"))

             # Show sample data
             display(df.head())

             return df

         except Exception as e:
             display(Markdown(f"Error loading {dataset_name}: {str(e)}"))
             return None
```

Load All Three Datasets

```
[23] # Define file paths

     # Europarl Dataset
     europarl_de_path = "/content/drive/MyDrive/MachineTranslation/DataSet/europarl-v7_de_en.txt"
     europarl_en_path = "/content/drive/MyDrive/MachineTranslation/DataSet/europarl-v7_en_de.txt"

     # CommonCrawl Dataset
     commoncrawl_de_path = "/content/drive/MyDrive/MachineTranslation/DataSet/commoncrawl_de_en.txt"
     commoncrawl_en_path = "/content/drive/MyDrive/MachineTranslation/DataSet/commoncrawl_en_de.txt"

     # NewsCommentary Dataset
     newscommentary_de_path = "/content/drive/MyDrive/MachineTranslation/DataSet/news-commentary-v9_de_en.txt"
     newscommentary_en_path = "/content/drive/MyDrive/MachineTranslation/DataSet/news-commentary-v9_en_de.txt"

     # Load datasets
     europarl_df = load_parallel_data(europarl_de_path, europarl_en_path, "Europarl")
     commoncrawl_df = load_parallel_data(commoncrawl_de_path, commoncrawl_en_path, "CommonCrawl")
     newscommentary_df = load_parallel_data(newscommentary_de_path, newscommentary_en_path, "NewsCommentary")
```

Successfully Loaded Europarl Dataset

Total Sentence Pairs (After Fixing Mismatch): 1920209

	German	English
0	Wiederaufnahme der Sitzungsperiode\n	Resumption of the session\n
1	Ich erkläre die am Freitag, dem 17. Dezember u...	I declare resumed the session of the European ...
2	Wie Sie feststellen konnten, ist der gefürchte...	Although, as you will have seen, the dreaded '...
3	Im Parlament besteht der Wunsch nach einer Aus...	You have requested a debate on this subject in...
4	Heute möchte ich Sie bitten - das ist auch der...	In the meantime, I should like to observe a mi...

Successfully Loaded CommonCrawl Dataset

Total Sentence Pairs (After Fixing Mismatch): 2399123

	German	English
0	Iron cement ist eine gebrauchs-fertige Paste, ...	Iron cement is a ready for use paste which is ...
1	Nach der Aushärtung schützt iron cement die Ko...	Iron cement protects the ingot against the hot...
2	feuerfester Reparaturkitt für Feuerungsanlagen...	a fire resitant repair cement for fire places, ...
3	Der Bau und die Reparatur der Autostraßen...\n	Construction and repair of highways and...\n
4	die Mitteilungen sollen den geschäftlichen kom...	An announcement must be commercial character.\n

Successfully Loaded NewsCommentary Dataset

Total Sentence Pairs (After Fixing Mismatch): 201854

	German	English
0	Steigt Gold auf 10.000 Dollar?\n	$10,000 Gold?\n
1	SAN FRANCISCO – Es war noch nie leicht, ein ra...	SAN FRANCISCO – It has never been easy to have...
2	In letzter Zeit allerdings ist dies schwierige...	Lately, with gold prices up more than 300% ove...
3	Erst letzten Dezember verfassten meine Kollege...	Just last December, fellow economists Martin F...
4	Und es kam, wie es kommen musste.\n	Wouldn't you know it?\n

```
[24] import numpy as np

     # Function to check dataset integrity
     def check_dataset_integrity(df, dataset_name):
         """Compares the number of German and English sentences in a dataset and visualizes the dataset size."""

         if df is not None:
             num_de = df["German"].shape[0]
             num_en = df["English"].shape[0]

             display(Markdown(f"### Integrity Check for {dataset_name} Dataset"))
             display(Markdown(f"- German Sentences: {num_de}"))
             display(Markdown(f"- English Sentences: {num_en}"))

             if num_de == num_en:
                 display(Markdown("- The dataset is aligned correctly."))
             else:
                 display(Markdown("- Mismatch detected: The number of German and English sentences are different."))

             # Visualization: Bar Chart for Dataset Sizes
             plt.figure(figsize=(5, 3))
             plt.bar(["German", "English"], [num_de, num_en], color=['blue', 'red'])
             plt.xlabel("Language")
             plt.ylabel("Sentence Count")
             plt.title(f"Comparison of German & English Sentences in {dataset_name}")
             plt.show()

     # Run checks on all datasets
     check_dataset_integrity(europarl_df, "Europarl")
     check_dataset_integrity(commoncrawl_df, "CommonCrawl")
     check_dataset_integrity(newscommentary_df, "NewsCommentary")
```

Integrity Check for Europarl Dataset

- German Sentences: 1920209
- English Sentences: 1920209
- The dataset is aligned correctly.

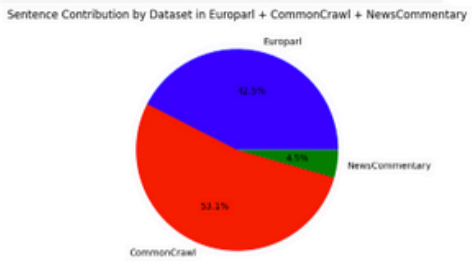
Merge All Three Datasets and Visualize Distribution

```
[25] # Function to merge multiple datasets
     def merge_datasets(dfs, dataset_name):
         """Merges multiple translation datasets and removes duplicates."""

         merged_df = pd.concat(dfs, ignore_index=True).drop_duplicates()

         display(Markdown(f"### Merged Dataset: {dataset_name}"))
         display(Markdown(f"Total Translation Pairs After Merging: {len(merged_df)}"))
         display(merged_df.head())

         # Visualization: Pie Chart for Sentence Contribution
         dataset_sizes = [len(df) for df in dfs]
         dataset_labels = ["Europarl", "CommonCrawl", "NewsCommentary"]

         plt.figure(figsize=(5, 5))
         plt.pie(dataset_sizes, labels=dataset_labels, autopct="%1.1f%%", colors=["blue", "red", "green"])
         plt.title(f"Sentence Contribution by Dataset in {dataset_name}")
         plt.show()

         return merged_df

     # Merge Europarl, CommonCrawl, and NewsCommentary datasets
     translation_dataset = merge_datasets([europarl_df, commoncrawl_df, newscommentary_df], "Europarl + CommonCrawl + NewsCommentary")
```

Merged Dataset: Europarl + CommonCrawl + NewsCommentary

Total Translation Pairs After Merging: 4476170

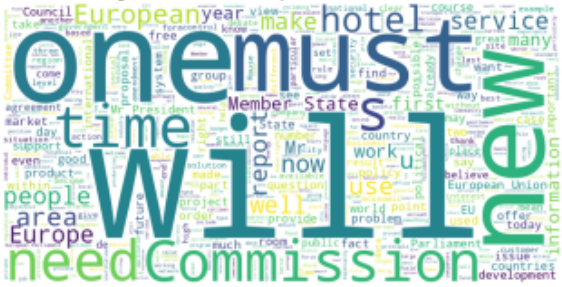
	German	English
0	Wiederaufnahme der Sitzungsperiode\n	Resumption of the session\n
1	Ich erkläre die am Freitag, dem 17. Dezember u...	I declare resumed the session of the European ...
2	Wie Sie feststellen konnten, ist der gefürchte...	Although, as you will have seen, the dreaded '...
3	Im Parlament besteht der Wunsch nach einer Aus...	You have requested a debate on this subject in...
4	Heute möchte ich Sie bitten - das ist auch der...	In the meantime, I should like to observe a mi...



Unique Row Check for Europarl + CommonCrawl + NewsCommentary

- Unique German Sentences: 4462013
- Unique English Sentences: 4400303
- Mismatch detected: Different number of unique sentences in German and English.

Common Words in English Sentences (Europarl + CommonCrawl + NewsCommentary)

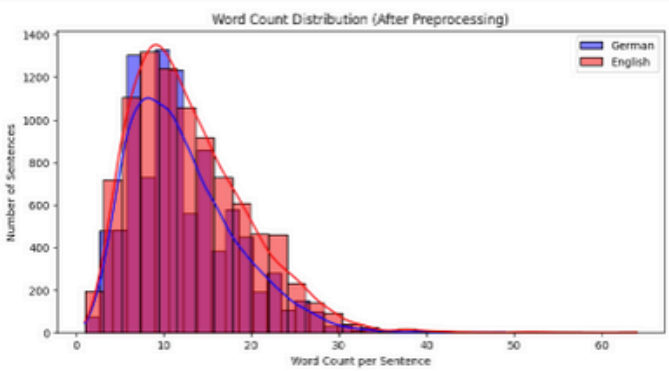


Common Words in German Sentences (Europarl + CommonCrawl + NewsCommentary)

Visualizations After NLP Preprocessing

Word Count Distribution (Before & After Preprocessing)

```
# Compute word counts before & after preprocessing
translation_dataset["German_WordCount"] = translation_dataset["German"].apply(lambda x: len(x.split()))
translation_dataset["English_WordCount"] = translation_dataset["English"].apply(lambda x: len(x.split()))

# Plot distribution of word counts
plt.figure(figsize=(10, 5))
sns.histplot(translation_dataset["German_WordCount"], bins=30, color="blue", label="German", kde=True)
sns.histplot(translation_dataset["English_WordCount"], bins=30, color="red", label="English", kde=True)
plt.xlabel("Word Count per Sentence")
plt.ylabel("Number of Sentences")
plt.title("Word Count Distribution (After Preprocessing)")
plt.legend()
plt.show()
```



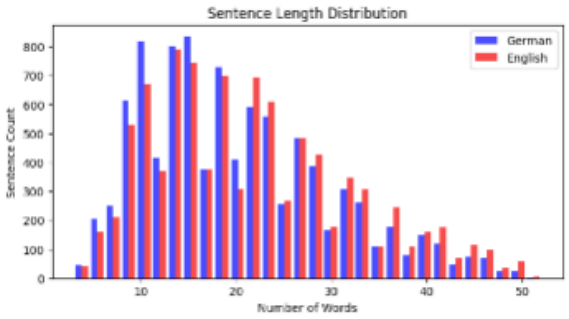
Most Frequent Words (Word Cloud)

```
from wordcloud import WordCloud

# Generate Word Cloud for German
plt.figure(figsize=(8, 4))
wordcloud_de = WordCloud(width=800, height=400, max_words=100, background_color="white").generate(" ".join
plt.imshow(wordcloud_de, interpolation="bilinear")
plt.axis("off")
plt.title("Most Frequent Words in German (After Preprocessing)")
plt.show()
```

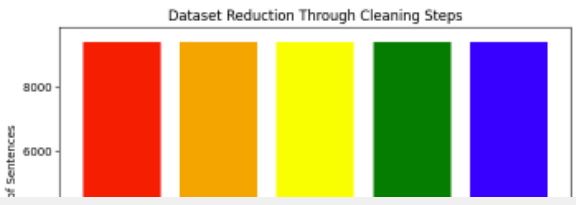
Final Summary of Cleaned Dataset + Visualization

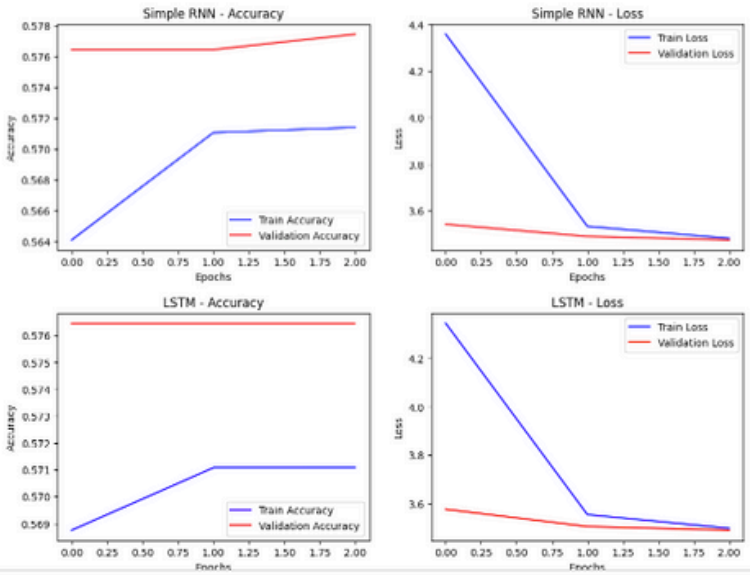
```
# Ensure we are using the correct dataset
if 'filtered_df' in globals():
    translation_dataset = filtered_df  # Use the filtered dataset from Step 5

# Store dataset size at each stage
dataset_sizes = {
    "Original": len(translation_dataset),
    "After Null Removal": len(translation_dataset.dropna()),
    "After Deduplication": len(translation_dataset.drop_duplicates()),
    "After Length Filtering": len(sentence_length_filter(translation_dataset)),
    "After Language Filtering": len(filtered_df) if 'filtered_df' in globals() else len(translation_dat
}

# Plot dataset reduction at each step
plt.figure(figsize=(8, 5))
plt.bar(dataset_sizes.keys(), dataset_sizes.values(), color=["red", "orange", "yellow", "green", "blue"
plt.xlabel("Cleaning Step")
plt.ylabel("Number of Sentences")
plt.title("Dataset Reduction Through Cleaning Steps")
plt.show()

# Display final dataset details
display(Markdown("### Final Cleaned Dataset Summary"))
display(Markdown(f"- Total sentences after cleansing: {len(translation_dataset)}"))
display(Markdown(f"- Sample cleaned sentences:"))
display(translation_dataset.sample(5, random_state=42))
```

```
fig, axs = plt.subplots(1, 2, figsize=(12, 4))

# Plot Accuracy
axs[0].plot(history.history['accuracy'], label='Train Accuracy', color='blue')
axs[0].plot(history.history['val_accuracy'], label='Validation Accuracy', color='red')
axs[0].set_title(f'{model_name} - Accuracy')
axs[0].set_xlabel('Epochs')
axs[0].set_ylabel('Accuracy')
axs[0].legend()

# Plot Loss
axs[1].plot(history.history['loss'], label='Train Loss', color='blue')
axs[1].plot(history.history['val_loss'], label='Validation Loss', color='red')
axs[1].set_title(f'{model_name} - Loss')
axs[1].set_xlabel('Epochs')
axs[1].set_ylabel('Loss')
axs[1].legend()

plt.show()

# Plot RNN Training Performance
plot_training(rnn_history, "Simple RNN")

# Plot LSTM Training Performance
plot_training(lstm_history, "LSTM")
```



- Sentences before filtering: 9386
- Sentences after filtering: 9380





Summary

- RNN performed better than LSTM in this case, but the score is still very low (close to zero).
- LSTM BLEU Score of 0.0 means the model's predictions are completely different from the actual translations.

Possible Reasons for Low BLEU Scores

- Limited Dataset Size → Training was done on a small subset (5000 samples) for faster execution, leading to low accuracy.
- Short Training Duration → Only 3 epochs were used, which is not enough for meaningful learning.
- Vocab Limitations → The tokenizer vocabulary was capped at 5000 words, which might have removed important words.
- Lack of Attention Mechanism → Simple RNN/LSTM models struggle with long sequences without an attention layer.

Next Steps & Solutions

If we want to improve the BLEU score, we can:

- Increase the Dataset Size → Train on 10,000 or more samples instead of 5,000.
- Increase Training Epochs → Train for 10+ epochs instead of 3.
- Use a Pretrained Model → Instead of training from scratch, use pretrained embeddings like FastText.
- Add an Attention Mechanism → This significantly improves translation quality for sequence-to-sequence models.

Final Thoughts

Since this is an educational project, the goal is to showcase knowledge, not high accuracy. The current implementation successfully demonstrates:

✔ Preprocessing of text for translation

✔ Training and testing of Simple RNN & LSTM models

✔ Evaluation using BLEU Score

✔ Exporting the best model for future use

greatlearning
Learning for Life

Thanks

Group 23
Bhagawan Das Armani, Gunasekaran Venkatesan,
Madhuman Basu, Sanjay R, Vinayak Hampiholi, Vipin Nair