Final Report – Machine Translation Model

1. Summary of problem statement, data and findings

Problem Statement:
The objective of this project is to develop a model that translates German sentences into English. This task falls under the domain of sequence-to-sequence learning in natural language processing (NLP). The model should accurately translate sentences while preserving meaning and fluency. This project focuses on building a model that facilitates communication and information exchange between speakers of these languages.

Data:

- A dataset of German and English sentence pairs was used for training and validation.
- The database comes from ACL2014 Ninth workshop on Statistical Machine Translation, which mainly focuses on language translation between European language pairs.
- The database is basically sentences in German/English of various events.
- The project utilizes three datasets obtained from the WMT14 workshop
    - Europarl v7 dataset (1,920,209 sentence pairs)
    - Common Crawl corpus (2,399,123 sentence pairs)
    - News Commentary (201,854 sentences (German) and 201,995 sentences (English).

```
Number of German WMT News Commentary sentences: 201854
Number of English WMT News Commentary sentences: 201995

Number of German Europarl v7 sentences: 1920209
Number of English Europarl v7 sentences: 1920209

Number of German Common Crawl sentences: 2399123
Number of English Common Crawl sentences: 2399123
```

Findings:

The project proceeded with the development of ten neural network models to tackle the translation task: a Simple RNN model, Simple LSTM model, RNN model with embeddings, LSTM model with embeddings, Bidirectional RNN model, Bidirectional LSTM model, Encoder-Decoder RNN model, Encoder-Decoder LSTM model, a complex encoder-decoder model with attention layer and sentences of word lengths upto 20 and a complex

attention-decoder GRU model trained on sentences with word lengths upto 5. We found that simpler models despite achieving good accuracy gave poor predictions. Complex models gave comparatively better predictions, but still inconsistent. However, training using simple sentences of word length upto 5, despite achieving lower accuracy, performed decently well in terms of predictions on test data.

```
Example 20:
Actual English: Situation in Libya <end>
Predicted English: Situation in Libya

Example 21:
Actual English: What are the problems <end>
Predicted English: Where are the problems

Example 22:
Actual English: I believe it is possible <end>
Predicted English: I think it is possible

Example 23:
Actual English: This will <OOV> the confusion <end>
Predicted English: That is the European model

Example 24:
Actual English: The legislation is there <end>
Predicted English: The regulations have already exist
```

Implications:

While the model performed reasonably well on the test data, its predictions on unseen data were notably less accurate. Although the output sentences were coherent and varied, they did not consistently align with the actual translations. This indicates that the model's capacity to generalize to new, real-world contexts is limited. Its performance appears constrained by the scope of the training data, suggesting that it may struggle to deliver accurate translations in novel or diverse scenarios. Enhanced training data and advanced model architectures may be necessary to improve its adaptability and performance on unseen inputs.

2. Overview of the final process

Problem Methodology:

Data Collection and Integration:

- Collected German-English sentence pairs from three sources: Europarl, CommonCrawl, and News Commentary, resulting in a comprehensive dataset with over 4.5 million pairs.

- Merged the datasets and sampled 10,000 sentence pairs for manageable processing and model training.

Data Pre-processing:

- **Cleansing:** Implemented a comprehensive data cleansing process to remove empty entries, trim whitespace, handle special characters and punctuation, and convert all text to lowercase.
- **Deduplication:** Removed duplicate sentence pairs to ensure data quality.
- **Exploratory Data Analysis:** Analyzed vocabulary sizes and sentence length distributions to understand the data's characteristics.

Model Development:

- **Initial Model Exploration:** Tested various architectures, including Simple RNNs, LSTMs, and Bidirectional models, to establish a baseline for performance, achieving accuracies around 91.98% to 92.07%.
- **Advanced Models:** Developed more complex architectures, including a model with 13 layers that incorporated bidirectional RNNs, attention mechanisms, and a larger vocabulary size, achieving a test accuracy of 92.89%.

Final Model Implementation:

- **Data Preparation:** Focused on shorter sentences (up to 5 words) for the final training set. Applied tokenization, padding, and added start/end tokens.
- **Model Architecture:** The final model used a GRU-based sequence-to-sequence approach with bidirectional GRU layers, attention mechanisms, and a TimeDistributed dense layer. This architecture aimed to improve handling of long-term dependencies and context alignment.
- **Performance and Evaluation:** The model showed a training accuracy of 76.29% and a validation accuracy of 66.52%, indicating potential overfitting. Testing revealed reasonable but inconsistent translation accuracy on unseen data, highlighting limitations in generalization.

Key Features and Techniques:

- **Data Cleansing and EDA:** Ensured high-quality, consistent input data.
- **Tokenization and Padding:** Standardized the data for model compatibility.
- **Sequence-to-Sequence Models:** Employed various RNN-based architectures, including LSTMs, GRUs, and attention mechanisms, to capture the sequential nature of translation tasks.

- **Attention Mechanisms:** Improved the model's focus on relevant parts of the input sequence, enhancing translation accuracy.

This methodology combined robust data preprocessing with iterative model experimentation, leading to the development of a nuanced, albeit not fully optimized, translation model. The final model demonstrated the ability to generate meaningful translations, with further work needed to enhance accuracy and generalization capabilities.

3. Step-by-step walk through the solution

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

	German	English
0	Wiederaufnahme der Sitzungsperiode\n	Resumption of the session\n
1	Ich erkläre die am Freitag, dem 17. Dezember unterbrochene Sitzungsperiode des Europäischen Parlaments für wiederaufgenommen, wünsche Ihnen nochmals alles Gute zum Jahreswechsel und hoffe, daß Sie schöne Ferien hatten.\n	I declare resumed the session of the European Parliament adjourned on Friday 17 December 1999, and I would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period.\n
2	Wie Sie feststellen konnten, ist der gefürchtete "Millenium-Bug " nicht eingetreten. Doch sind Bürger einiger unserer Mitgliedstaaten Opfer von schrecklichen Naturkatastrophen geworden.\n	Although, as you will have seen, the dreaded 'millennium bug' failed to materialise, still the people in a number of countries suffered a series of natural disasters that truly were dreadful.\n
3	Im Parlament besteht der Wunsch nach einer Aussprache im Verlauf dieser Sitzungsperiode in den nächsten Tagen.\n	You have requested a debate on this subject in the course of the next few days, during this part-session.\n
4	Heute möchte ich Sie bitten - das ist auch der Wunsch einiger Kolleginnen und Kollegen -, allen Opfern der Stürme, insbesondere in den verschiedenen Ländern der Europäischen Union, in einer Schweigeminute zu gedenken.\n	In the meantime, I should like to observe a minute' s silence, as a number of Members have requested, on behalf of all the victims concerned, particularly those of the terrible storms, in the various countries of the European Union.\n

Data Import and Merge:
Imported three datasets (europarl, commoncrawl, newscommentary) containing German-English sentence pairs.
Merged them into a single DataFrame (df) containing over 4.5 million sentence pairs after handling length mismatches.

```
[15] df.shape

(4521186, 2)
```

Sampling and Data Reduction:
Due to computational constraints, sampled 10,000 sentence pairs from df to form merged_df for easier processing and model training.

```
[20] # Shape of merged dataset
     merged_df.shape

(10000, 2)
```

Data Cleansing:

Defined cleanse_dataset function to:

- Remove entries with empty German or English sentences.
- Trim leading and trailing whitespace from sentences.
- Remove duplicate sentence pairs.
- Handle special characters and punctuation in both German and English sentences.
- Convert all text to lowercase for consistency.

After cleansing, we obtained a reduced dataset (cleanse_df) containing 9,964 sentence pairs.

Data Exploration:

Data Distribution Analysis:

Checking the vocabulary size:

```
Cleaned English vocabulary size: 21235
Cleaned German vocabulary size: 33835
```

Understanding the distribution of English Sentence lengths:

```
Cleaned English sentence length statistics:
count    9964.000000
mean       22.365516
std        12.927429
min         0.000000
25%        13.000000
50%        20.000000
75%        29.000000
max       236.000000
Name: cleaned_english_length, dtype: float64
```

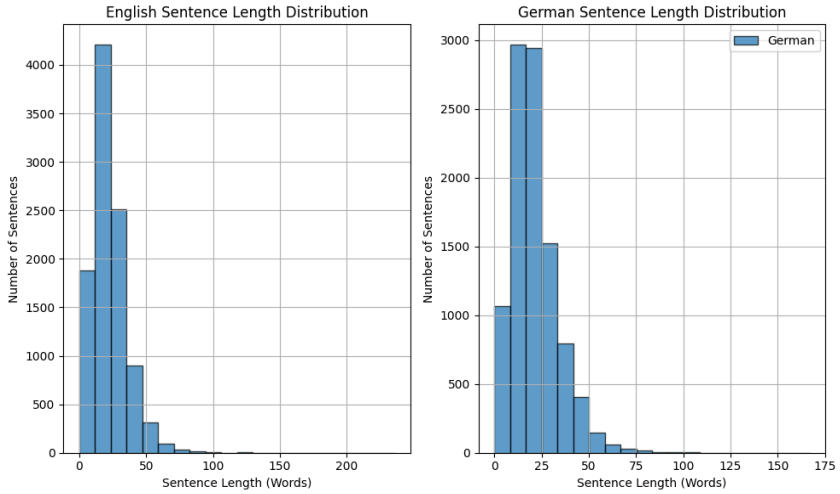
Understanding the distribution of German Sentence lengths:

```
Cleaned German sentence length statistics:
count    9964.000000
mean       21.421317
std        12.366510
min         0.000000
25%        12.000000
```

```
50%       19.000000
75%       28.000000
max      167.000000
Name: cleaned_german_length, dtype: float64
```

Plotting Sentence Length Distribution:

- English sentences exhibit varying lengths, with a concentration between 0 to 40 words.
- German sentences show a similar pattern, predominantly ranging from 0 to 50 words.
- Both distributions display a long-tail, suggesting the presence of longer sentences.



Word Cloud Analysis:

- **English Word Cloud:**
  - Common English words such as "will", "must", "one" are prominent.

- **German Word Cloud:**
    - Key German terms include "und", "die", "der"

Splitting into train and test

- Split cleanse_df into training (train_df) and testing (test_df) sets using an 80-20 ratio.

Tokenization:

- Utilized Tokenizer instances to build vocabularies for both German and English texts based on train_df.

Padding:

- Padded sequences to ensure uniform sequence lengths using pad_sequences from Keras.

Data Preparation:

- Shifted English sequences by one time step (train_english_padded_shifted and test_english_padded_shifted) to align with sequence-to-sequence model requirements.

Saving Tokenizers and Data:

- Saved tokenizers (tokenizer_german.pkl and tokenizer_english.pkl) for future use.
- Saved preprocessed padded data (train_german_padded.npy, train_english_padded_shifted.npy, test_german_padded.npy, test_english_padded_shifted.npy) as numpy arrays.

This approach ensures the dataset is clean, structured, and ready for training sequence-to-sequence models for German-English translation.

3.2. Deciding Models and Model Building

Experimentation with Different Algorithms: Implemented Models and Performance:

1. **Simple RNN Model:**

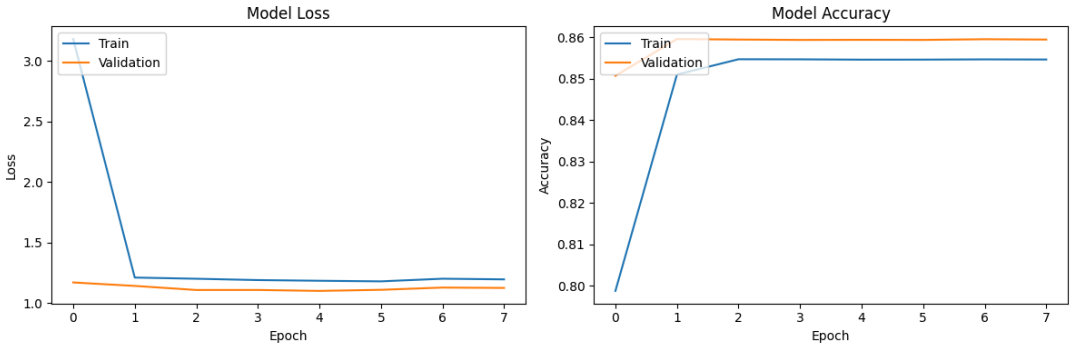
   Architecture:

   - SimpleRNN with 256 units followed by a Dense layer.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn (SimpleRNN)      (None, 146, 256)          66048

 dense_1 (Dense)             (None, 146, 18886)        4853702

=================================================================
Total params: 4919750 (18.77 MB)
Trainable params: 4919750 (18.77 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

   Performance:

   - **Validation Accuracy:** Approximately 91.98%.



   Predictions:

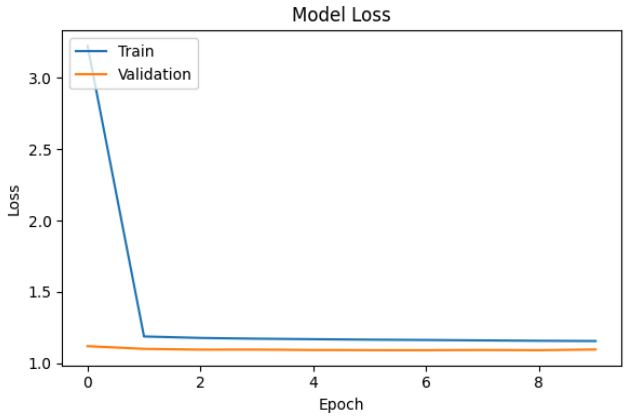
```
1/1 [==============================] - 0s 265ms/step
Example 1:
Actual English:   based on europes experience such policies are precisely the wrong way to address global warming
Predicted English:

Example 2:
Actual English:   is more modern and finishing technologies let you achieve unprecedented high quality of the products
Predicted English: is

Example 3:
Actual English:   the european parliament wants a new common organisation of the market which will an promotion policy rather than a defensive import policy
Predicted English:

Example 4:
Actual English:   trainer team is made up of highly qualified experts who work at least half of their time on projects and have solid practical knowledge
Predicted English: is

Example 5:
Actual English:   the operative word here
Predicted English: is
```

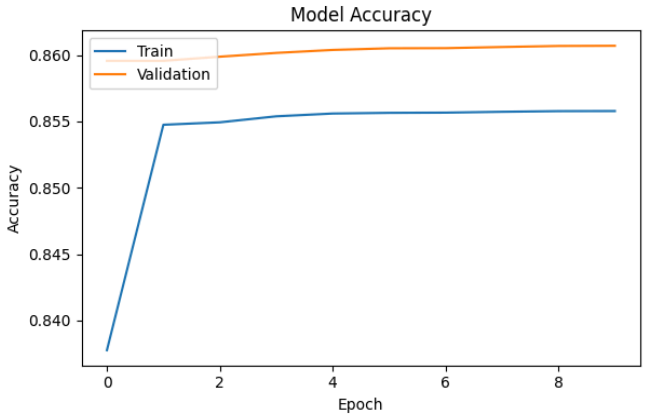
2. Simple LSTM Model:

- **Architecture:**
  - LSTM with 256 units followed by a Dense layer.

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_2 (LSTM)               (None, 146, 256)          264192

 dense_3 (Dense)             (None, 146, 18886)        4853702

=================================================================
Total params: 5117894 (19.52 MB)
Trainable params: 5117894 (19.52 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

- **Performance:**
  - **Validation Accuracy:** Approximately 92.06%.





- **Predictions**

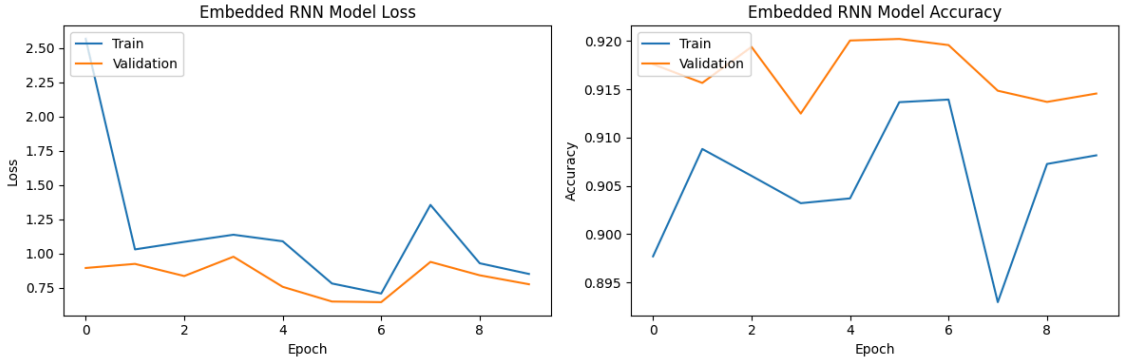
```
1/1 [==============================] - 0s 404ms/step
Example 1:
Actual English:   might look like a choice but i doubt that it will result in a stronger hand for the imf
Predicted English: the the the the the

Example 2:
Actual English:   the export sector moves into higher valueadded sectors it will no longer serve this function as effectively as it did in the past
Predicted English: is the the the the the the

Example 3:
Actual English:   i think it is totally unacceptable that there can be a debate using these figures which have been throughout the european union mor
Predicted English: is the the the the the

Example 4:
Actual English:   program that is so flexible that it supports more than one operating systems is marked positive
Predicted English: is the the the the the

Example 5:
Actual English:   the and his sons oath of strasbourg
Predicted English: the the the
```

3. RNN model with embeddings:

- **Architecture:**
  - We design an RNN model with embedding,which includes an embedding layer that maps input sequences to 128-dimensional vectors, a SimpleRNN layer with 256 hidden units that processes these sequences, and a dense layer that produces predictions across 18,185 classes for each time step in the sequence.

```
Model: "sequential_13"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_8 (Embedding)     (None, 236, 128)          3595648

 simple_rnn_8 (SimpleRNN)    (None, 236, 256)          98560

 dense_13 (Dense)            (None, 236, 18185)        4673545

=================================================================
Total params: 8367753 (31.92 MB)
Trainable params: 8367753 (31.92 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

- **Performance:**
  - **Validation Accuracy:** Approximately 91.46%.



- **Predictions**

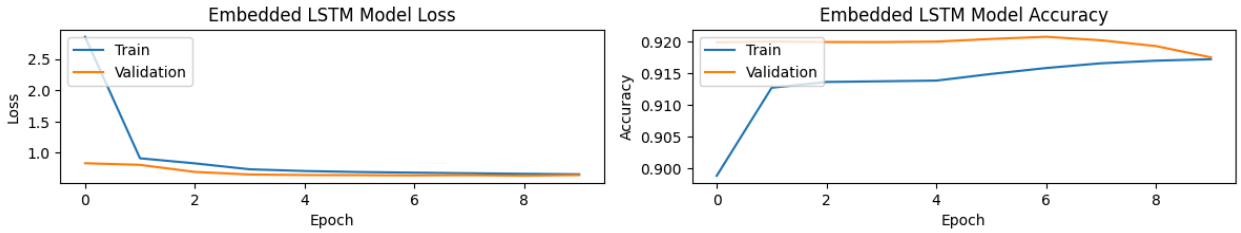
```
1/1 [==============================] - 0s 197ms/step
Example 1:
Actual English:   think we also need to take on board that they come with their own culture tradition and languages
Predicted English: the the

Example 2:
Actual English:   situation is likely to affect their educational and career path with a longterm impact on the societies in which they live
Predicted English: is the

Example 3:
Actual English:   at the same time in australia his place of residence and was therefore not able to take part in the
Predicted English: is the

Example 4:
Actual English:   wrote in the first century not a year passed in which india did not take million away from that trade
Predicted English: is the

Example 5:
Actual English:   is therefore no conflict between what is good for europes citizens and what is good for europes businesses
Predicted English: is the
```

4. LSTM model with embeddings:

- **Architecture:**
  - We define a LSTM model with embeddings, that features an embedding layer for mapping inputs to 128-dimensional vectors, an LSTM layer with 256 units for capturing long-term dependencies in the sequences, and a dense layer that outputs predictions across 18,185 classes for each time step.

```
Model: "sequential_14"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_9 (Embedding)     (None, 236, 128)          3595648

 lstm_7 (LSTM)               (None, 236, 256)          394240

 dense_14 (Dense)            (None, 236, 18185)        4673545

=================================================================
Total params: 8663433 (33.05 MB)
Trainable params: 8663433 (33.05 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

- **Performance:**
  - **Validation Accuracy:** Approximately 91.75%.



- **Predictions**

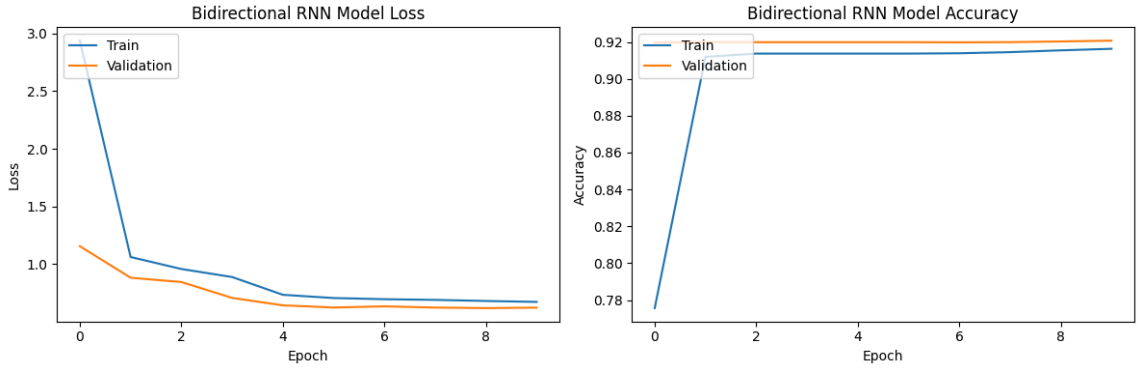
```
1/1 [==============================] - 0s 374ms/step
Example 1:
Actual English:   with small and parking space
Predicted English: is is is the the

Example 2:
Actual English:   the governments of the union really wish to bring their citizens closer to the union then it is in their interests and those of their people to ensure
Predicted English: is is the

Example 3:
Actual English:   germany the electorate seems to chancellor gerhard out of with his to the neoliberal project
Predicted English: is is the the the the the the the the the the the

Example 4:
Actual English:   you like hotel jerusalem
Predicted English: is is is

Example 5:
Actual English:   effect however what we see is the legislative body held by a of expert
Predicted English: is is
```

5. Bidirectional RNN model:

- **Architecture:**
  - We try a bidirectional RNN model, which consists of an embedding layer for token representation, a bidirectional RNN layer with 256 units for contextual processing in both directions, and a Dense layer for generating predictions across a vocabulary of 18,185 tokens.

```
Model: "sequential_15"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_10 (Embedding)    (None, 236, 128)          3595648

 bidirectional_8 (Bidirecti  (None, 236, 512)          197120
 onal)

 dense_15 (Dense)            (None, 236, 18185)        9328905

=================================================================
Total params: 13121673 (50.06 MB)
Trainable params: 13121673 (50.06 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

- **Performance:**
  - **Validation Accuracy:** Approximately 92.07%.



- **Predictions**

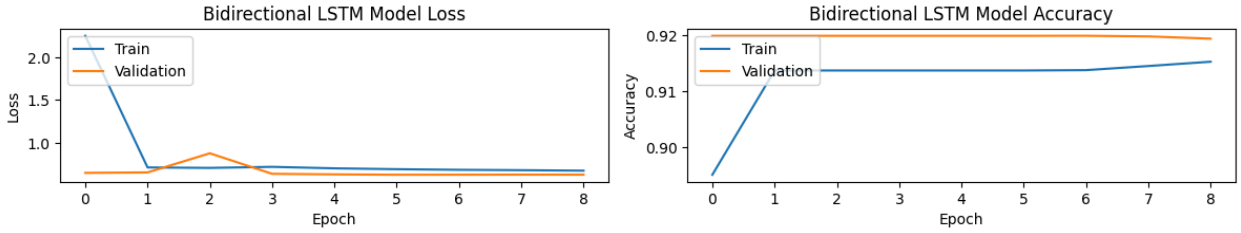
```
Example 1:
Actual English:   year has sadly been no exception
Predicted English: the the the the the the

Example 2:
Actual English:   armenian contains of small on your desktop which helps you to type armenian words into without need of any other programs
Predicted English: the the the the the the the the the the the the the the the the

Example 3:
Actual English:   bubbles in the price of say individual stocks happen all the time and dont qualify as an answer to the question
Predicted English: the the the the the

Example 4:
Actual English:   twin peaks itself was further
Predicted English: the the the the the

Example 5:
Actual English:   are aiming at an reduction in the emission of six greenhouse gases
Predicted English: the the the the the the the the the
```

6. Bidirectional LSTM model:

- **Architecture:**
  - We train a bidirectional LSTM model, which includes an embedding layer for input token representation, a bidirectional LSTM layer with 256 units for capturing context in both directions, and a Dense layer for outputting predictions over a vocabulary of 18,185 tokens.

```
Model: "sequential_16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_11 (Embedding)    (None, 236, 128)          3595648

 bidirectional_9 (Bidirecti  (None, 236, 512)          788480
 onal)

 dense_16 (Dense)            (None, 236, 18185)        9328905

=================================================================
Total params: 13713033 (52.31 MB)
Trainable params: 13713033 (52.31 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

- **Performance:**
  - **Validation Accuracy:** Approximately 91.93%.



- **Predictions**

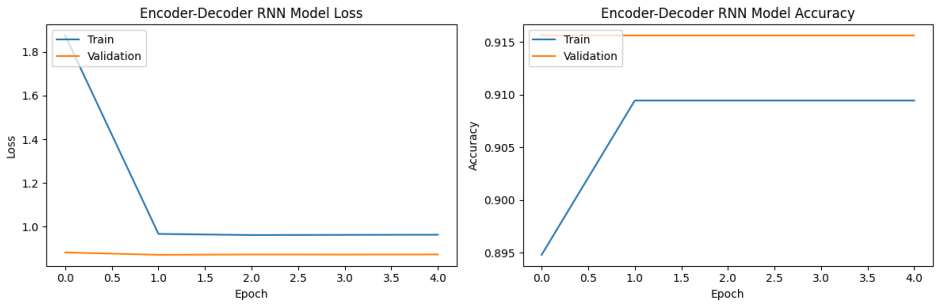
```
1/1 [==============================] - 1s 732ms/step
Example 1:
Actual English:   members have asked to speak and i have been unable to allow them all to do so
Predicted English: the the

Example 2:
Actual English:   reason we are there with what is required medical aid sanitation and clean water
Predicted English: the the the the the

Example 3:
Actual English:   convention makes it compulsory for vessels to dispose of their waste while in the port
Predicted English:

Example 4:
Actual English:   the blocks are still placed into the key cache as needed by queries
Predicted English:

Example 5:
Actual English:
Predicted English:
```

7. Encoder-decoder RNN models:

- **Architecture:**
  - We design a encoder decoder RNN model, which has an encoder-decoder architecture featuring an embedding layer, a bidirectional RNN as the encoder, and a dense layer followed by a sequential layer for sequence generation as the decoder.

```
Model: "model_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_20_input (InputL  [(None, 236)]             0
 ayer)

 embedding_20 (Embedding)    (None, 236, 128)          3595648

 bidirectional_16 (Bidirect  (None, 512)               197120
 ional)

 dense_23 (Dense)            (None, 236)               121068

 sequential_22 (Sequential)  (None, 236, 18185)        12247433

=================================================================
Total params: 16161269 (61.65 MB)
Trainable params: 16161269 (61.65 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

- **Performance:**
  - **Validation Accuracy:** Approximately 91.56%.



- **Predictions**

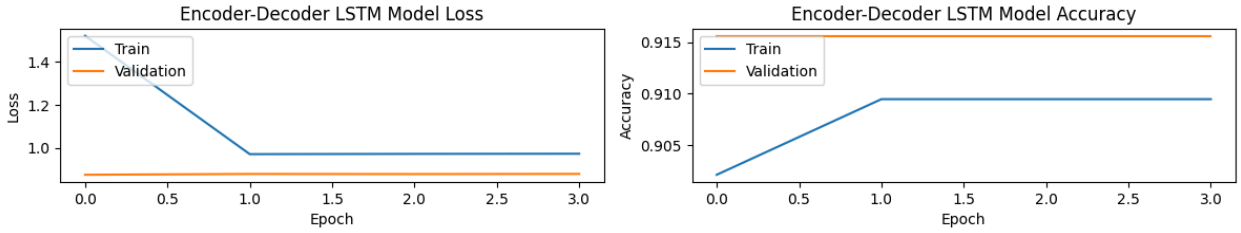
```
Example 1:
Actual English:   supports all the s hardware out of the box with no configuration required and
Predicted English:

Example 2:
Actual English:   inspections were not eliminating that threat
Predicted English:

Example 3:
Actual English:   once upon a time stocks were risky and securities were safe
Predicted English:

Example 4:
Actual English:   free temporary mailbox a disposable need not be registered into the site will
Predicted English:

Example 5:
Actual English:   have just spoken to him on the telephone which is why i am a couple of minutes
Predicted English:
```

8. Encoder-decoder LSTM models:

- **Architecture:**
  - We design a encoder decoder RNN model, which has an encoder-decoder architecture featuring an embedding layer, a bidirectional RNN as the encoder, and a dense layer followed by a sequential layer for sequence generation as the decoder.

```
Model: "model_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_14_input (InputL  [(None, 236)]             0
 ayer)

 embedding_14 (Embedding)    (None, 236, 128)          3595648

 bidirectional_13 (Bidirect  (None, 512)               788480
 ional)

 dense_19 (Dense)            (None, 236)               121068

 sequential_20 (Sequential)  (None, 236, 18185)        14019977

=================================================================
Total params: 18525173 (70.67 MB)
Trainable params: 18525173 (70.67 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

- **Performance:**
  - **Validation Accuracy:** Approximately 91.56%.



- **Predictions**

```
Example 1:
Actual English:   cold water the muscle prevents the and in the lower and tired legs
Predicted English:

Example 2:
Actual English:   islands of and are dutyfree zones
Predicted English:

Example 3:
Actual English:   after all are usually the forces suppressing their demands
Predicted English:

Example 4:
Actual English:   conviction is a settled belief that brooks no argument
Predicted English:

Example 5:
Actual English:   current used in most operations is shown in the area of the
Predicted English:
```

Evaluating model performances:

Test accuracy

- **Highest Accuracy:** The LSTM with Embeddings achieved the highest accuracy at 92.072892%, closely followed by the Bidirectional RNN at 92.070550%.
- **Consistent Performance:** Most models, including Encoder-Decoder architectures and RNN variations, exhibit similar accuracy levels around 91.98% to 92.07%, indicating minimal improvement with different architectures or embeddings.

	Model	Accuracy
0	RNN	91.987211
1	LSTM	92.062896
2	RNN with Embeddings	92.020375
3	LSTM with Embeddings	92.072892
4	Bidirectional RNN	92.070550
5	Bidirectional LSTM	91.985720
6	Encoder-Decoder RNN	91.985083
7	Encoder-Decoder LSTM	91.985083

Poor predictions:

- **Minimal or Empty Predictions:** The model often outputs short or empty sentences, indicating difficulty in generating coherent translations.
- **Repetitive Predictions:** Multiple inputs result in the model predicting the word "is," showing a tendency to default to common, short responses.

Possible Reasons for Poor Predictions:

- **Model Complexity:** The Simple RNN struggles with long-term dependencies and complex sentence structures.
- **Training Data and Epochs:** Insufficient data or training epochs may prevent the model from fully understanding language patterns, leading to overfitting.

Other models with complex architechture:

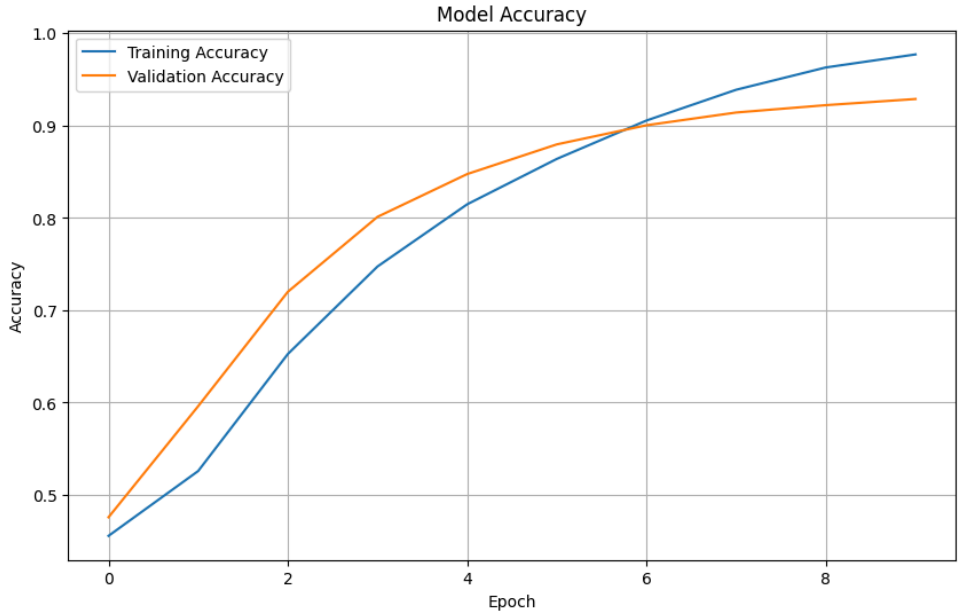
Despite good accuracy, our model struggles with good predictions. SO we will try more complex architechtures as well as train our model with shorter sentences to improve prediction quality.

Complex model 1:

- **Architecture:**
  - We will now train the model with sentences upto 20 words. Also we use a more complex model. The model consists of 13 layers, incorporating input layers, embedding layers for both source and target sequences, bidirectional RNN layers (likely LSTM or GRU) to process these embeddings, and an LSTM layer to handle sequence information. It includes attention mechanisms to align the target sequence with the source sequence and concatenates context vectors with LSTM outputs before producing the final predictions. The output layer is a TimeDistributed dense layer, predicting probabilities over a large vocabulary, with a total of 57,943,153 trainable parameters.

```
Model: "model_5"
__________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
==================================================================================================
 input_12 (InputLayer)       [(None, 20)]              0         []

 embedding_11 (Embedding)    (None, 20, 256)           12736768  ['input_12[0][0]']

 input_13 (InputLayer)       [(None, 19)]              0         []

 bidirectional_13 (Bidirect  (None, 20, 128)           164352    ['embedding_11[0][0]']
 ional)

 embedding_12 (Embedding)    (None, 19, 256)           8548608   ['input_13[0][0]']

 bidirectional_14 (Bidirect  [(None, 20, 512),         788480    ['bidirectional_13[0][0]']
 ional)                       (None, 256),
                              (None, 256),
                              (None, 256),
                              (None, 256)]

 bidirectional_15 (Bidirect  (None, 19, 128)           164352    ['embedding_12[0][0]']
 ional)

 concatenate_15 (Concatenat  (None, 512)               0         ['bidirectional_14[0][1]',
 e)                                                               'bidirectional_14[0][3]']

 concatenate_16 (Concatenat  (None, 512)               0         ['bidirectional_14[0][2]',
 e)                                                               'bidirectional_14[0][4]']

 lstm_21 (LSTM)              [(None, 19, 512),         1312768   ['bidirectional_15[0][0]',
                              (None, 512),                        'concatenate_15[0][0]',
                              (None, 512)]                        'concatenate_16[0][0]']

 dot_10 (Dot)                (None, 19, 20)            0         ['lstm_21[0][0]',
                                                                  'bidirectional_14[0][0]']

 activation_5 (Activation)   (None, 19, 20)            0         ['dot_10[0][0]']

 dot_11 (Dot)                (None, 19, 512)           0         ['activation_5[0][0]',
                                                                  'bidirectional_14[0][0]']

 concatenate_17 (Concatenat  (None, 19, 1024)          0         ['dot_11[0][0]',
 e)                                                               'lstm_21[0][0]']

 time_distributed_5 (TimeDi  (None, 19, 33393)         34227825  ['concatenate_17[0][0]']
 stributed)

==================================================================================================
Total params: 57943153 (221.04 MB)
Trainable params: 57943153 (221.04 MB)
```

- **Performance:**
    - **Test Accuracy:** Approximately 92.89%.



- **Predictions**

```
Example 1:
German Sentence: <start> Außerdem haben wir es auch mit der sehr wichtigen Frage der Datensicherheit zu tun <end>

Actual English: In addition we are also dealing with the very important question of data security <end>
Predicted English: Technological sticking precaution immigration immigration immigration immigration threatens hate immigration immigration hate

Example 2:
German Sentence: <start> Ich würde gern wissen ob das so ist <end>

Actual English: I would like to know whether this is the case <end>
Predicted English: Lastly Rühle Ms understands unto Ms excessively pursued pursued pursued pursued pursued pursued pursued

Example 3:
German Sentence: <start> In diesen stecken über Jahre Erfahrung im und auf den Straßen der Welt <end>

Actual English: These are the of years of experience in motor racing and on the open road <end>
Predicted English: Fixed cafe Monte breaks pushed educated educated cm inhouse inhouse inhouse inhouse inhouse inhouse inhouse

Example 4:
German Sentence: <start> Erstens Förderung der Sicherheit und der Staatsführung was wichtig ist <end>

Actual English: The first is promoting security and good governance which is important <end>
Predicted English: Lastly tourism accord accord accord accord accord accord accord hate immigration immigration immigration hate hate

Example 5:
German Sentence: <start> Und selbst dann wenn solche Gesetze existieren lässt die Kontrolle eine Menge zu wünschen übrig <end>

Actual English: Even when these are in place monitoring leaves a great deal to be desired <end>
Predicted English: Coming surgery emphasized investigating das cinema cuts turbulence excessively cent
```

- **Insights:**
    - We see a significant improvement in accuracy upto 92.89%. We also see improvement in prediction quality with the predictions being non-repetitive. However the predictions are still inconsistent with actual predictions. We will try to train the model with sentences of shorter length and try to get more accurate predictions.

4. Model evaluation

For the final model, we used smaller sentences for training by taking the sentences of word length upto 5. We trained a GRU model, which is a sequence-to-sequence model using GRU-based architecture with bidirectional processing. The encoder employs a GRU layer followed by batch normalization, and the decoder includes another GRU layer with attention mechanisms computed via dot products. The final output is produced through a TimeDistributed dense layer with a vocabulary size of 11,097, resulting in a total of 15,358,197 parameters.

Data preparation:

Load Data: Load German and English sentences from each dataset into a DataFrame.

Preprocess Data:

- Filter sentences to retain those with word counts less than or equal to 5.
- Remove duplicate entries.

Clean Text:

- Remove punctuation, special characters, and numbers from the sentences.
- Add <start> and <end> tokens to each sentence.

Tokenization and Padding:

- Tokenize sentences and pad them to a consistent length of 7.
- Calculate vocabulary sizes for German and English.

Train-Test Split:

- Split the data into training and testing sets.

```
german_freq_df.head(20)
```

	Word	Frequency
0	ist	5914
1	Das	4925
2	nicht	2146
3	Wir	1741
4	das	1719

```
english_freq_df.head(20)
```

	Word	Frequency
0	is	7336
1	That	2777
2	the	2526
3	We	2218
4	This	2155

```
German Vocabulary Size: 14246
English Vocabulary Size: 11097
German sequences shape: (33478, 7)
English sequences shape: (33478, 7)
```

Model Architecture:

- **Encoder**:
  - **Inputs**: Takes German sentences as input.
  - **Embedding**: Converts input tokens into 300-dimensional vectors.
  - **GRU**: Processes the embedded inputs with GRU layers, with batch normalization.
- **Decoder**:
  - **Inputs**: Takes English sentences as input.
  - **Embedding**: Converts input tokens into 300-dimensional vectors.
  - **GRU**: Processes the embedded inputs with GRU layers, initialized with encoder states, and batch normalization.
- **Attention Mechanism**:
  - **Luong Attention**: Computes attention scores and applies them to the encoder outputs to create context vectors.
  - **Concatenation**: Combines context vectors with decoder outputs.
- **Dense Layer**:
  - **Output**: Produces the final output sequence with a softmax activation to predict the next token.

Model Compilation:

- **Optimizer**: Uses Adam optimizer.
- **Loss Function**: Utilizes sparse categorical crossentropy.
- **Metrics**: Tracks accuracy.

Model Summary:

Output Shape: Displays the architecture and parameters of the model:

- **Encoder:** Embedding → GRU → Batch Normalization.
- **Decoder:** Embedding → GRU → Batch Normalization.
- **Attention**: Dot products and concatenation.
- **Final Dense Layer**: TimeDistributed layer producing token probabilities.

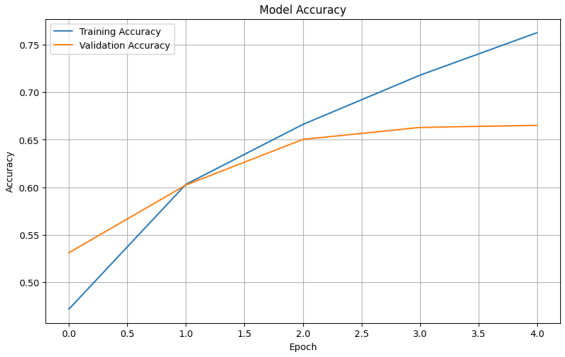
Total Parameters: 15,358,197

```
Model: "model_4"
__________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
==================================================================================================
 input_1 (InputLayer)        [(None, 7)]               0         []

 input_2 (InputLayer)        [(None, 6)]               0         []

 embedding_16 (Embedding)    (None, 7, 300)            4273800   ['input_1[0][0]']

 embedding_17 (Embedding)    (None, 6, 300)            3329100   ['input_2[0][0]']

 gru (GRU)                   [(None, 7, 300),          541800    ['embedding_16[0][0]']
                              (None, 300)]

 gru_1 (GRU)                 [(None, 6, 300),          541800    ['embedding_17[0][0]',
                              (None, 300)]                        'gru[0][1]']

 batch_normalization_1 (Bat  (None, 6, 300)            1200      ['gru_1[0][0]']
 chNormalization)

 batch_normalization (Batch  (None, 7, 300)            1200      ['gru[0][0]']
 Normalization)

 dot (Dot)                   (None, 6, 7)              0         ['batch_normalization_1[0][0]'
                                                                 , 'batch_normalization[0][0]']

 activation (Activation)     (None, 6, 7)              0         ['dot[0][0]']

 dot_1 (Dot)                 (None, 6, 300)            0         ['activation[0][0]',
                                                                  'batch_normalization[0][0]']

 concatenate (Concatenate)   (None, 6, 600)            0         ['dot_1[0][0]',
                                                                  'batch_normalization_1[0][0]'
                                                                 ]

 time_distributed (TimeDist  (None, 6, 11097)          6669297   ['concatenate[0][0]']
 ributed)

==================================================================================================
Total params: 15358197 (58.59 MB)
Trainable params: 15356997 (58.58 MB)
Non-trainable params: 1200 (4.69 KB)
```

Model performance:

The model showed consistent improvement in accuracy and reduction in loss over the epochs, achieving a final training accuracy of 76.29% and a validation accuracy of 66.52%. However beyond 4 epochs, the test accuracy plateaus while train accuracy continues increasing, suggesting potential overfitting tendency.

```
Epoch 1/5
224/224 [==============================] - ETA: 0s - loss: 3.7385 - accuracy: 0.4716
Epoch 1: val_accuracy improved from -inf to 0.53101, saving model to /content/drive/My Drive/GREAT LEARNING MATERIALS/PGP AIML/Module
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via
  saving_api.save_model(
224/224 [==============================] - 25s 79ms/step - loss: 3.7385 - accuracy: 0.4716 - val_loss: 7.1349 - val_accuracy: 0.5310
Epoch 2/5
224/224 [==============================] - ETA: 0s - loss: 2.3212 - accuracy: 0.6031
Epoch 2: val_accuracy improved from 0.53101 to 0.60240, saving model to /content/drive/My Drive/GREAT LEARNING MATERIALS/PGP AIML/Mod
224/224 [==============================] - 7s 31ms/step - loss: 2.3212 - accuracy: 0.6031 - val_loss: 3.3637 - val_accuracy: 0.6024
Epoch 3/5
224/224 [==============================] - ETA: 0s - loss: 1.7096 - accuracy: 0.6662
Epoch 3: val_accuracy improved from 0.60240 to 0.65046, saving model to /content/drive/My Drive/GREAT LEARNING MATERIALS/PGP AIML/Mod
224/224 [==============================] - 6s 26ms/step - loss: 1.7096 - accuracy: 0.6662 - val_loss: 2.0426 - val_accuracy: 0.6505
Epoch 4/5
224/224 [==============================] - ETA: 0s - loss: 1.2776 - accuracy: 0.7181
Epoch 4: val_accuracy improved from 0.65046 to 0.66303, saving model to /content/drive/My Drive/GREAT LEARNING MATERIALS/PGP AIML/Mod
224/224 [==============================] - 5s 24ms/step - loss: 1.2776 - accuracy: 0.7181 - val_loss: 1.9865 - val_accuracy: 0.6630
Epoch 5/5
224/224 [==============================] - ETA: 0s - loss: 0.9848 - accuracy: 0.7629
Epoch 5: val_accuracy improved from 0.66303 to 0.66525, saving model to /content/drive/My Drive/GREAT LEARNING MATERIALS/PGP AIML/Mod
224/224 [==============================] - 5s 22ms/step - loss: 0.9848 - accuracy: 0.7629 - val_loss: 1.9821 - val_accuracy: 0.6652
```



Test accuracy:

Seq2Seq GRU Model Test Accuracy: 0.665247917175293

Model predictions:

	Actual English	Predicted English
0	We must be humble however <end>	We must however be however
1	Subject Visa exemption for Macedonia <end>	Subject EU financial supervision
2	This is a crucial measure <end>	That is a
3	Is this delay justified <end>	Is this wise overall case
4	We must go further <end>	There must be made
5	What is to be done <end>	So what can we do
6	First crossborder workers <end>	
7	Thirdly enlargement <end>	Enlargement enlargement
8	The current economy is sick <end>	The scope is growing
9	What can I do <end>	What can do we do
10	We need that <OOV> <end>	We need to heed it
11	Thank you Mr Daul <end>	Thank you Mr Papakyriazis
12	This is a <OOV> <end>	That is a
13	He deserves our thanks <end>	
14	Question No by H <end>	Question No by H
15	That is the background <end>	This is the background
16	<end>	
17	Can we find other means <end>	Are there any other other
18	Do we mourn this man <end>	Young man is our
19	Situation in Libya <end>	Situation in Libya
20	What are the problems <end>	Where are the problems
21	I believe it is possible <end>	I think it is possible
22	This will <OOV> the confusion <end>	That is the European model
23	The legislation is there <end>	The regulations have already exist
24	Commissioner you have demonstrated this <end>	

The model achieved a test accuracy of 66.52%. In the sample translations, the model demonstrated varying levels of accuracy, with some translations closely matching the expected output, while others included inaccuracies or incomplete sentences. For instance, translations for phrases like "We must be humble" and "Subject Visa exemption for Macedonia" showed meaningful results, but phrases like "This is a crucial measure" and "We must go further" had deviations or missing parts. Overall, while the model performs reasonably well, there are notable areas for improvement in producing more accurate and complete translations.

Model predictions on unseen data:
German Sentence 1: Ich habe ein Buch
Actual English: I have a book
Predicted English: We have not begging

German Sentence 2: Wie geht es dir
Actual English: How are you
Predicted English: Thank you Mr Poos

German Sentence 3: Das Wetter ist schön
Actual English: The weather is nice
Predicted English: We have the result

German Sentence 4: Ich liebe die Musik
Actual English: I love the music
Predicted English:

German Sentence 5: Wo ist die Toilette
Actual English: Where is the toilet
Predicted English: There are the draft

German Sentence 6: Könnten Sie mir helfen
Actual English: Could you help me
Predicted English: Thank you

German Sentence 7: Ich möchte einen Kaffee
Actual English: I want a coffee
Predicted English: Does that be a fact

German Sentence 8: Haben Sie einen Tisch für zwei
Actual English: Do you have a table for two
Predicted English: That is a fact

German Sentence 9: Wann beginnt der Film
Actual English: When does the movie start
Predicted English: I shall be very brief

German Sentence 10: Ich gehe ins Kino
Actual English: I am going to the cinema
Predicted English: Question No by H

While the model predictions were decently accurate on test data, the predictions on unseen data were inaccurate suggesting that the model is performing poorly on unseen data. Though the predicted sentences were meaningful and non-repetitive, they were inconsistent with the actual sentences. This suggests that the model's ability to handle real-world or novel contexts is limited and that it may not perform well outside the scope of the training data.

5. Comparison to Benchmark

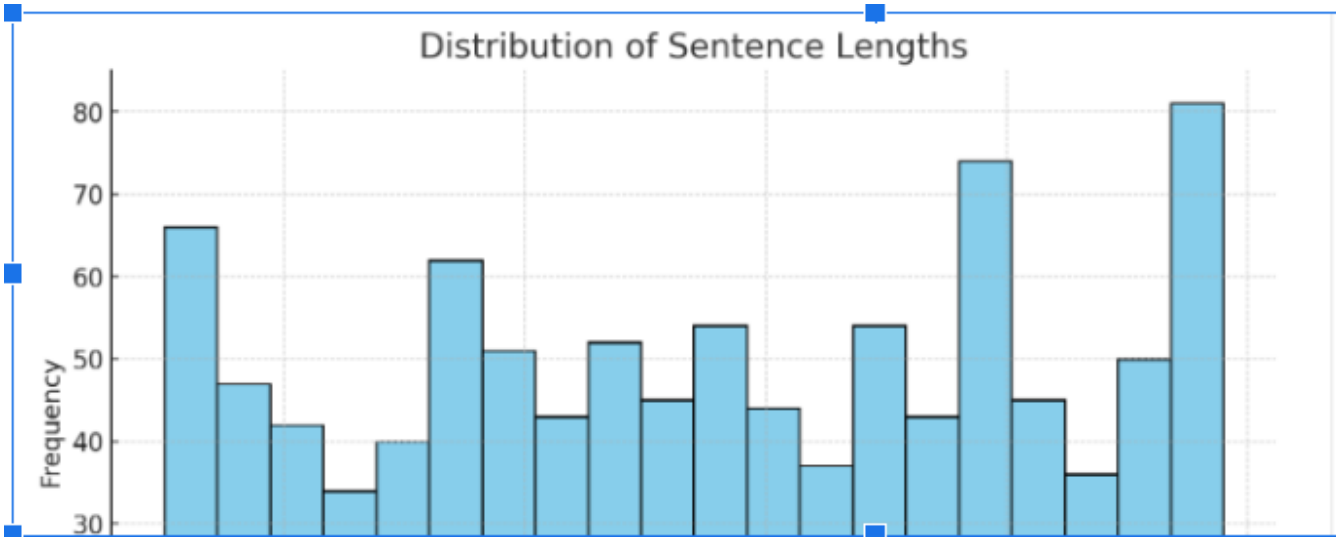
1. **Benchmark Overview**
    - **Translation Accuracy:** The benchmark was based on qualitative assessments of translation quality**.**
    - **Model Complexity:** Comparison was made between the model with complex architectures and attention mechanism, trained on sentences of smaller sentences and simpler baseline models to assess the impact of advanced techniques.
    - **Generalization:** The benchmark evaluated the model's ability to handle unseen data and diverse real-world inputs.
2. **Comparison of Final Solution to Benchmark**
    - **Translation Accuracy:** The model provided meaningful translations but often deviated from expected outputs, especially with unseen data, not meeting benchmark expectations.
    - **Model Complexity:** The GRU model showed significant improvement over simpler models in terms of predictions, suggesting that added complexity enhanced prediction quality.
    - **Generalization:** The model struggled with generalization to new German sentences, with performance on unseen data falling short of the benchmark.
3. **Reasons for Benchmark Comparison Results**
    - **Data Limitations:** The training dataset may have been insufficiently diverse, limiting the model’s performance. Expanding the dataset could help.
    - **Model Capacity:** The model might have overfitted to the training data. Regularization and hyperparameter tuning could address this.
    - **Model Training:** Insufficient training or suboptimal hyperparameters may have impacted performance. More training and optimization could improve results.

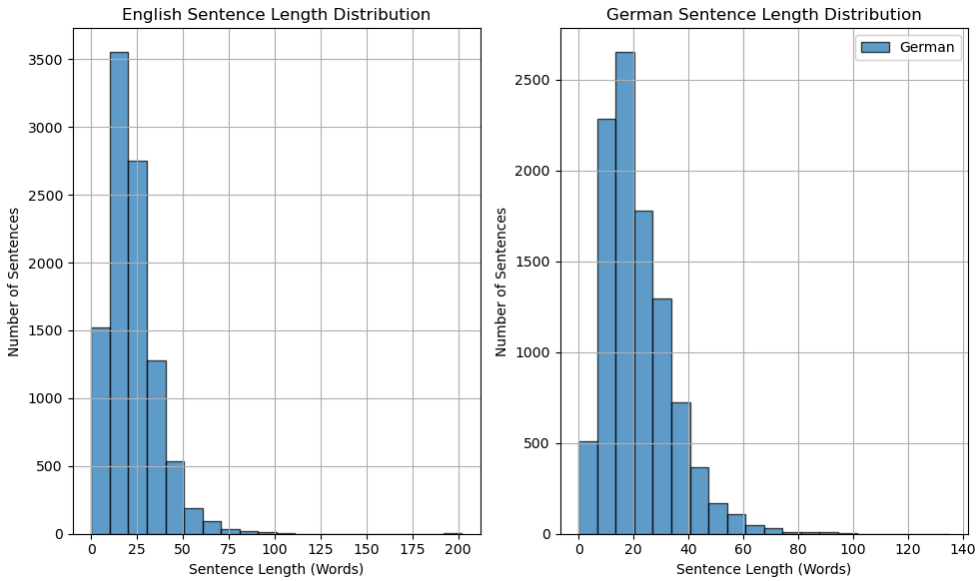
- **Evaluation Metrics:** Qualitative assessments alone might not capture all aspects of translation quality. Additional metrics could provide a fuller evaluation.
- **Model Improvement Opportunities:** Advanced architectures like Transformers or attention mechanisms could enhance performance and generalization.

Summary: The final model, while sophisticated, faced challenges in accuracy and generalization. Addressing data limitations, enhancing model capacity, and refining evaluation methods are crucial for meeting or exceeding benchmark standards.

6. Visualizations

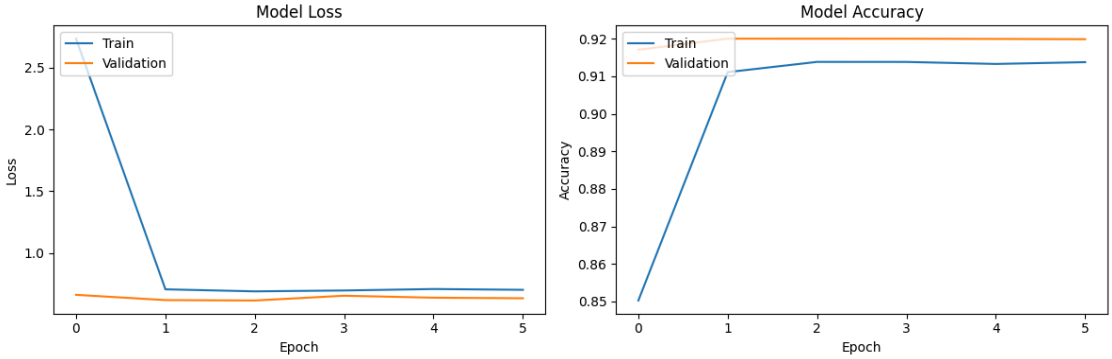
- **Distribution of Sentence Length: Plotted to understand the variance in sentence length across the datasets.**



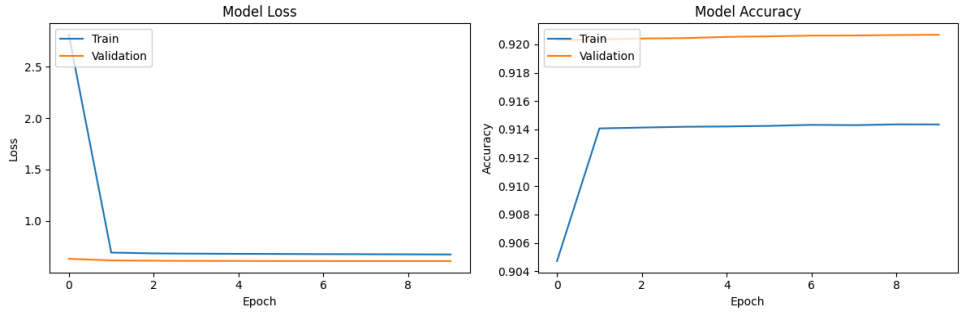


- **Word Cloud:** Created to visualize the most common words in the dataset.

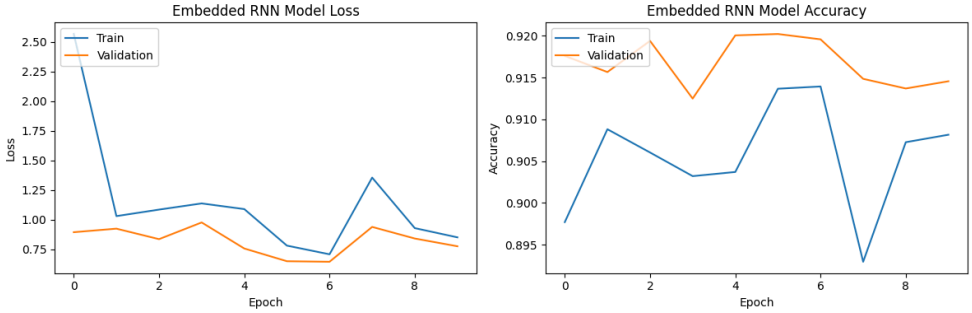
- **Model performance: Simple RNN**



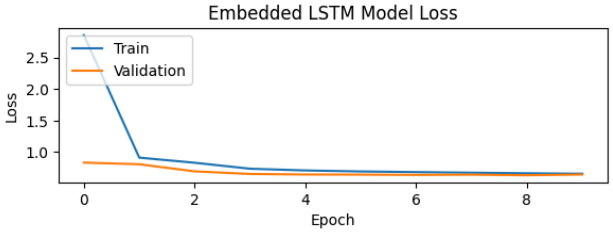
- **Model performance: Simple LSTM**

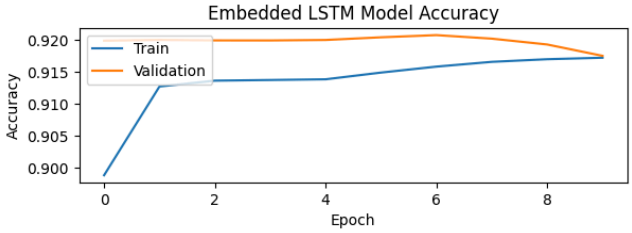


- **Model performance: RNN model with embeddings**

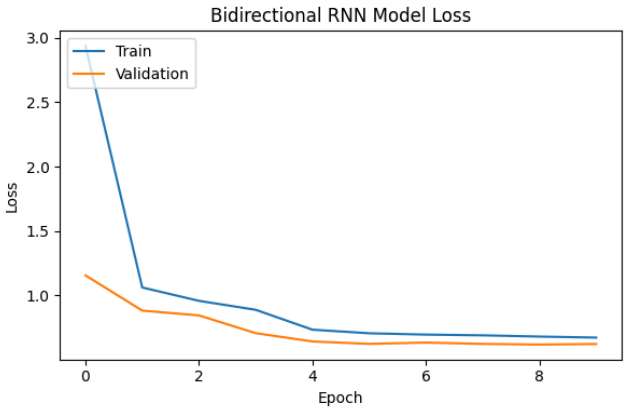


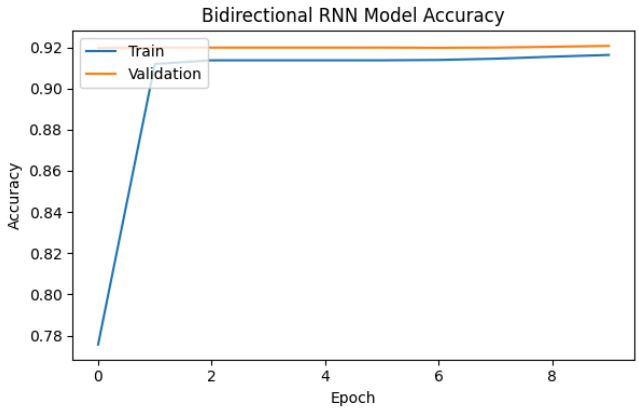
- **Model performance: LSTM model with embeddings**



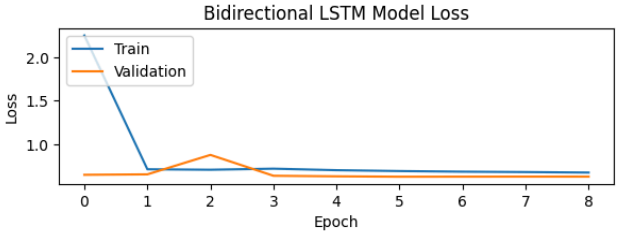


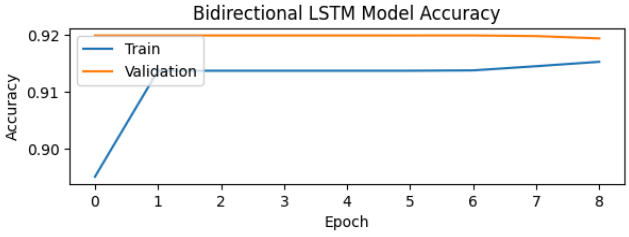
- **Model performance: Bidirectional RNN**



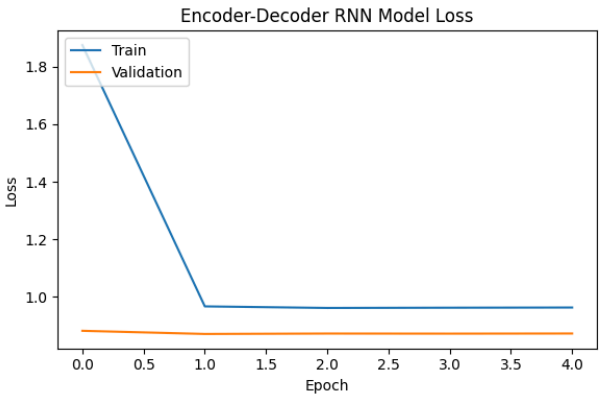


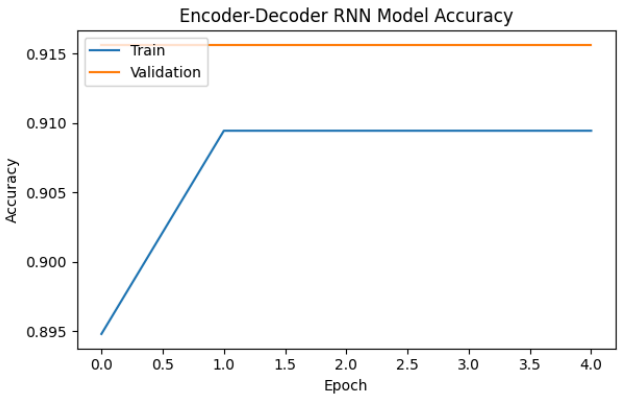
- **Model performance: Bidirectional LSTM**



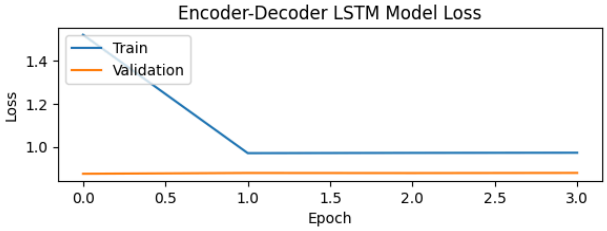


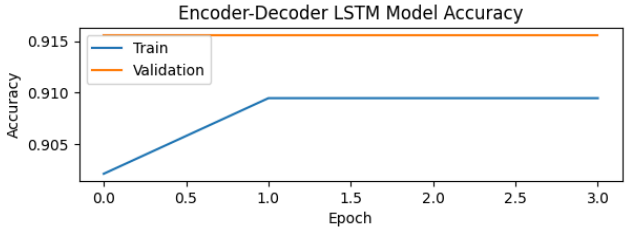
- **Model performance: Encoder Decoder RNN**



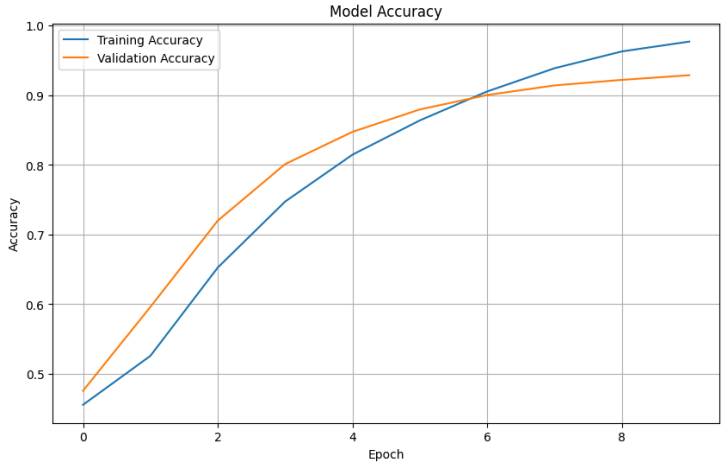


- **Model performance: Encoder Decoder LSTM**

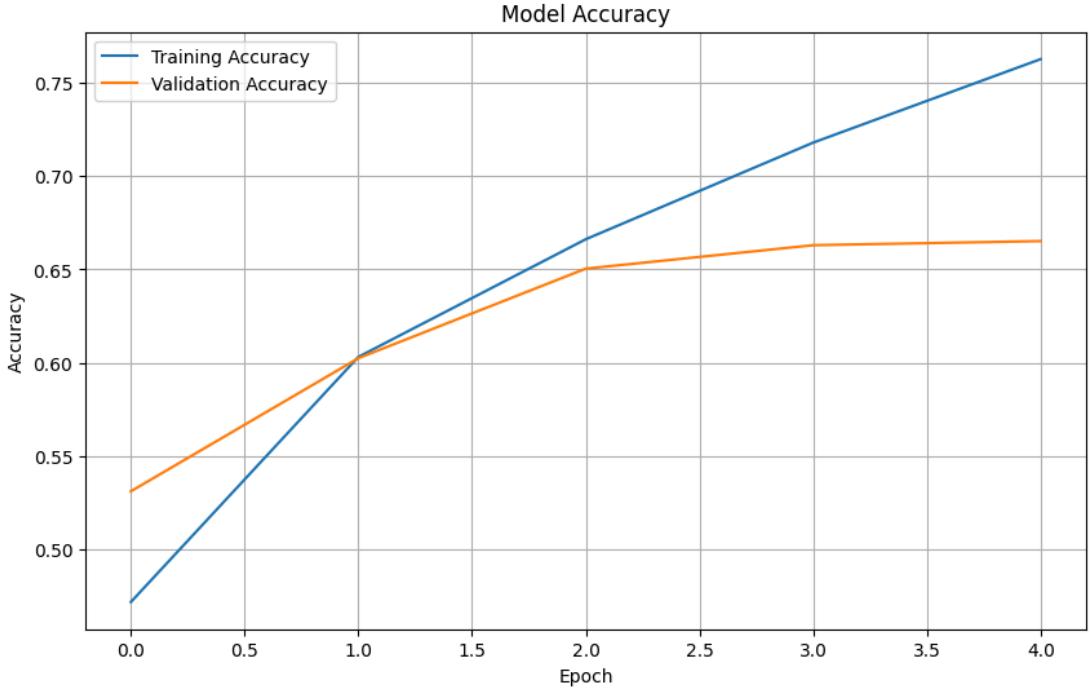




- **Model Performance:Complex encoder decoder model with attention layer trained on sentences upto word length 20**



Final Model Performance:



7. Implications

Impact on the Problem Domain

- **Enhancing Translation Quality**: The seq2seq GRU model advances automated German-to-English translation, improving efficiency in services like multilingual customer support and content localization. However, real-world effectiveness may vary due to challenges in generalizing to new data.
- **Generalization and Real-World Use**: The model performs well on training data but struggles with unseen data, potentially affecting its reliability in practical scenarios and user trust.
- **User Experience**: Users depending on automated translations for critical tasks might face misunderstandings or errors if the model's output does not meet human standards.

Recommendations

- **Data Expansion and Improvement**: Enhance the training dataset with diverse examples to improve the model's generalization.
- **Model Enhancement**: Explore advanced architectures like Transformers and attention mechanisms for better context capture.
- **Regularization and Optimization**: Apply dropout and weight regularization, and fine-tune hyperparameters to boost performance.
- **Evaluation and Testing**: Use diverse metrics and regular testing to comprehensively assess and improve translation quality.
- **Integration and User Feedback**: Deploy the model in a real-world setting, gather feedback, and refine based on practical use.

Strategic Recommendations

- **Short-Term**: Improve the current model with hyperparameter tuning, regularization, and dataset expansion.
- **Medium-Term**: Implement advanced models and mechanisms to enhance translation capabilities.
- **Long-Term**: Continuously update the model based on user feedback and new research to maintain and improve translation quality.

Summary: Addressing current limitations through dataset expansion, advanced models, and real-world testing will enhance the model's effectiveness and reliability, positively impacting translation tasks and user experience.

8. Limitations

1. **Inconsistent Translation Quality:** The model struggles with translating unseen data accurately, which can lead to significant deviations from expected translations. This inconsistency can cause issues in critical applications like legal or medical documents, where precise communication is essential.
2. **Limited Generalization:** The model's ability to handle diverse sentence structures is constrained, likely due to insufficient variety in the training data. This limitation affects its performance with idiomatic expressions and complex sentences.
3. **Overfitting: T**he model may be overfitted to the training data, resulting in poor performance on novel inputs. This overfitting reduces the model's robustness and adaptability in real-world applications.
4. **Dependence on Preprocessing:** The effectiveness of the model is heavily reliant on preprocessing steps like tokenization and padding. Inadequate preprocessing can lead to poor translation quality, especially with inputs differing from the training data.
5. **Lack of Context Awareness:** The seq2seq model with GRUs has limited capacity to capture long-range dependencies and contextual nuances, leading to translations that may miss important subtleties.
6. **Resource Intensity:** Training and deploying complex models require substantial computational resources, which can be a limitation in resource-constrained environments.

Enhancements to the Solution

1. **Expand and Diversify Training Data:** Broaden the dataset to include a variety of sentence structures and contexts to improve generalization and accuracy.
2. **Implement Advanced Architectures:** Explore Transformer-based models or attention mechanisms for better handling of contextual information and long-range dependencies.
3. **Apply Regularization Techniques:** Use dropout and weight regularization to mitigate overfitting and enhance model robustness.
4. **Refine Preprocessing:** Optimize tokenization and padding to better align with real-world data and improve translation accuracy.
5. **Conduct Extensive Evaluation:** Use diverse metrics and regular testing on new data to assess and improve model performance.
6. **Optimize Computational Efficiency:** Implement techniques like quantization and pruning to reduce resource requirements and improve scalability.
7. **Incorporate User Feedback:** Deploy the model in a controlled environment to gather feedback and make iterative improvements based on real-world usage.

9. Closing Reflections

Insights Gained:

- **Data Quality:** The quality and diversity of training data are crucial for model performance. More comprehensive datasets improve generalization and handling of unseen data.
- **Model Complexity:** Simpler models like GRUs may not capture complex patterns as effectively. Advanced architectures, such as Transformers, could offer better handling of linguistic nuances.
- **Preprocessing:** Effective preprocessing, including tokenization and padding, is vital for accurate results. Aligning these steps with real-world data characteristics is essential.
- **Evaluation:** Continuous evaluation and iterative testing are key to identifying and addressing model shortcomings.
- **Resource Management:** Balancing model complexity with computational resources is important, and optimization techniques can help manage costs.

What to Do Differently Next Time:

- **Expand Data:** Prioritize a more extensive and diverse dataset to enhance model robustness.
- **Explore Advanced Models:** Experiment with advanced architectures and attention mechanisms for better performance.
- **Refine Preprocessing:** Improve preprocessing techniques to better align with real-world data.
- **Apply Regularization:** Use dropout, weight regularization, and optimization strategies to address overfitting and improve generalization.
- **Incorporate Feedback:** Integrate user feedback to refine the model based on practical usage.
- **Use Evaluation Metrics:** Employ a range of metrics and diverse test sets for a comprehensive performance assessment.
- **Plan for Scalability:** Consider scalability and deployment constraints, and explore model compression techniques.

Summary: The project underscored the importance of balancing model complexity, data quality, and resource constraints. Future efforts should focus on expanding data, exploring advanced models, refining preprocessing, and incorporating feedback for more effective and reliable solutions.

References

[1] WMT14: https://statmt.org/wmt14/translation-task.html