ECE 1508 Final Project: Text Translation using YouTube Video Transcripts (Group 33)

Jiaxiang E
Department of ECE
University of Toronto
`jiaxiang.e@mail.utoronto.ca`

Yixu Ye
Department of ECE
University of Toronto
`yixu.ye@mail.utoronto.ca`

Shuqi Yang
Department of ECE
University of Toronto
`shuqiyang.yang@mail.utoronto.ca`

Abstract

This project aims at achieving English to French translation with a focus on deep learning methods, beginning with Recurrent Neural Network (RNN) and advancing to Sequence-to-Sequence (Seq2Seq) models, encompassing transformers. By capitalizing on the extensive bilingual content from the YouTube Video Subtitles API, the initiative employs a thorough multi-stage preprocessing regimen post-data extraction, which includes data cleansing, tokenization, and semantic enrichment through Word2Vec, coupled with VecMap for semantic space alignment across languages. This careful, iterative approach involves rolling out diverse models, fine-tuning pre-existing models, and conducting comparative assessments to enhance translation precision. By melding cutting-edge neural network frameworks with complex natural language processing tactics, the project aspires to craft highly accurate translation models.

Assentation of Teamwork

Jiaxiang E: Data preprocessing, basic RNN model, Seq2Seq transformer and report.

Yixu Ye: Data extraction, GRU model and LSTM RNN model Seq2Seq GRU model and LSTM RNN model and report.

Shuqi Yang: Word2Vec, VacMap RNN model and report.

Introduction

As internationalization deepens, the challenge of language translation has become increasingly vital for effective communication in various spheres of life. Nowhere is this more evident than in Canada, where the country boasts two official languages: English and French. The project aims to develop deep learning models that translate English to French by learning YouTube's translated text while maintaining high accuracy and contextual integrity. The deep learning model's ability to predict word sequences, understand idiomatic expressions, and adapt to the structural nuances of English and French allows it to identify regular patterns in large and broad datasets, rather than being bound by various grammatical rules as traditional human translators. A crucial aspect is that deep learning models are effective at managing long-range dependencies encountered in linguistic data. The reason behind this is their internal structures and mechanisms, such as attention in neural networks, that allow them to navigate the complexity of language with extraordinary skill.

Solution Derivative

Our project adopts a layered approach to language translation, starting with the extraction of bilingual transcripts from YouTube to assemble a rich vocabulary across a wide array of subjects. The initial stage involves purifying this data through noise removal and tokenization, followed by applying Word2Vec to transform words into semantic vectors. Subsequently, VecMap was employed to synchronize these vectors across English and French within a shared semantic framework, setting a solid foundation for our bilingual translation model.

Focusing on model development, we explore a variety of RNN architectures, including Vanilla RNN, LSTM (Long Short-Term Memory), and Gated Recurrent Unit (GRU). Despite the limitations of Vanilla RNNs, particularly in handling long-term dependencies,[1] the advancements in LSTM and GRU models have significantly improved the capture of textual context, proving invaluable for translation accuracy. While Transformers and BERT models currently dominate NLP benchmarks, RNNs retain their relevance due to their simplicity and effectiveness in specific scenarios[2].[3]

Our objective is to develop a deep neural network that excels in translation by evaluating diverse architectures, notably Seq2Seq, where one RNN functions as the encoder of the source text and another as the decoder into the target language. Through careful model selection and fine-tuning, we aim to enhance translation fidelity. The performance of our models is meticulously assessed using metrics like Bilingual Evaluation Study (BLEU), sentence cosine similarity, and Metric for Evaluation of Translation with Explicit Ordering (METEOR), with the goal of continuously refining translation quality. Notably, an increase in BLEU scores has been observed with the expansion of the training dataset. We also monitor Negative Log Likelihood (NLL) loss across iterations to ensure progressive improvement.

Inspired by groundbreaking work in the field, including RNN models achieving 94% accuracy[4] and an LSTM-based model by Zack Thoutt with 97% validation accuracy,[5] our project, while more straightforward, integrates these insights to guide our methodology. The end goal is to craft a deep learning translation system that not only evaluates the efficacy of traditional models against advanced technologies like Transformers but also enriches our understanding of RNNs' capabilities and limitations in the evolving landscape of natural language processing.

Design and Implement

Data Extraction

First of all, the process starts with extracting YouTube video transcripts via the `https://github.com/jdepoix/youtube-transcript-api` library to build our bilingual database. This initial step forms the foundation of our bilingual database, enriching our dataset with diverse linguistic characteristics found in educational content, interviews, and other documentaries.

Data Cleaning and Preprocessing

After retrieving the transcript data from YouTube, the project embarked on a nuanced preprocessing journey to refine the dataset for translation tasks. Initially, a broad spectrum of video categories was chosen to ensure lexical diversity, with the dataset segmented into 80% for training, 10% for validation, and 10% for testing purposes. Subsequent steps involve tokenizing the textual content and utilizing Word2Vec models, sourced from both English and a specialized French repository (found at `https://fauconnier.github.io/#data`), to convert the text into semantic vectors. Additionally, VecMap (details at `https://github.com/artetxem/vecmap/blob/master/README.md`) is employed to align these vectors into a cohesive semantic space, thereby refining the translation model's precision.

The process includes establishing English and French vocabularies, linking each word to a corresponding token within their respective languages, as exemplified in the token list figureFigure 1. Frequencies of word occurrences were calculated based on YouTube transcripts, alongside the removal of punctuation and atypical symbols. The resulting vocabulary sizes amounted to 4,334 for English and 5,051 for French(see Figure 5 in appendix). Analysis of sentence length distributions prompted the exclusion of sentences shorter than five or longer than twenty words, alongside their French counterparts, to maintain quality and relevance.

The sentences were then dissected into individual words and tokenized, incorporating special tokens for the start (**<SOS>**) and end (**<EOS>**) of sentences, where indicated by 1 and 2 after tokenize. These text data, once tokenized, were padded to match the maximum sequence length identified across the samples, ensuring uniformity for model ingestion. The Word2Vec model was leveraged to compute vector representations for each word, facilitating the evaluation of semantic similarities essential for the learning methodologies section's later stages. The maximum sequence length is defined to be the maximum sequence length among the existing sentence samples. Each word's vector is computed by word2vec model to obtain the similarities (see Figure 6 in appendix) for later evaluation in the methods training model section.

```
English vocabulary size: 4334
French vocabulary size: 5051
First English sentence: when the winter world disappears, and the sun's rays begin to gather strength, it's the start of a season that is a law unto itself
First English sentence tokens: [1, 867, 1199, 3, 1200, 868, 8, 27, 1926, 5, 869, 1927, 2]
First French sentence: quand le monde hivernal disparaît, et les rayons du soleil commencer à reprendre des forces, c'est le début d'une saison c'est une loi en soi
First French sentence tokens: [1, 15, 687, 8, 1225, 6, 855, 2028, 140, 4, 45, 2029, 9, 2030, 2031, 2]
```

Figure 1: Sample output of vocabularies and tokens.

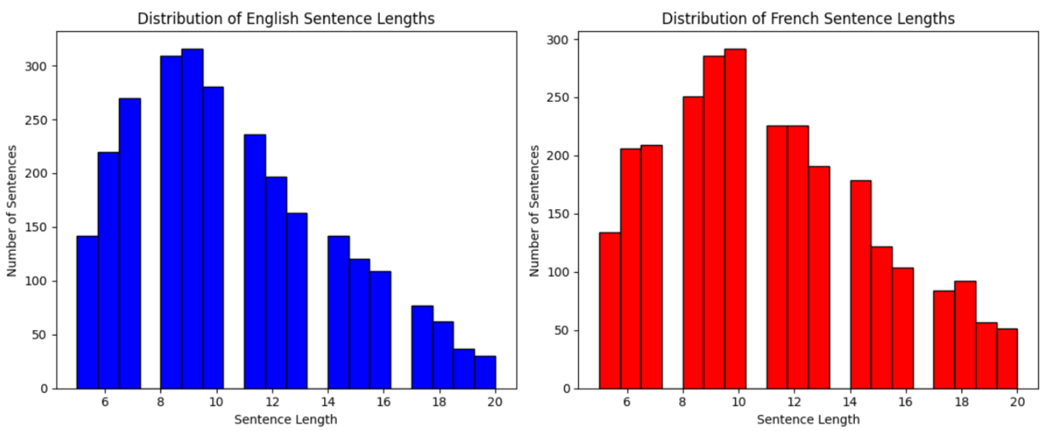


Figure 2: Sequence length histograms after filtering.

Model Design

Hardware Aspect

The NVIDIA A100 GPU is employed which is available in Google Colab.[6] This is a high-end GPU launched by NVIDIA for data centers and high-performance computing (HPC), and belongs to the Ampere architecture series.[7] The powerful computing power of the A100 GPU is very suitable for deep learning tasks. A100 has Tensor Core, a processing core specially optimized for deep learning operations, which can greatly accelerate matrix operations, shorten model training time, and increase the iteration speed of experiments.

Basic RNN

Recurrent Neural Networks are one of the most basic deep learning architectures crafted to interpret and process sequential data input, which is also known as the "vanilla" RNN. The mathematics formula is defined to be: $h_t = \tanh(W_{ih} \cdot x_t + b_{ih} + W_{hh} \cdot h_{t-1} + b_{hh})$.

The model generates a corresponding sequential data output. The model we've constructed adheres to the following configurations:

1. **Embedding layer:**
   The first embedding layer is the initial processing step that receives input sequences with English vocabulary size. It converts these input token indices into dense vectors of size 256. This transformation facilitates the learning of token representations that capture semantic meanings.

2. **RNN layer:**

   The recurrent unit layer in this model processes the embedded input sequences. It has an input size of 256, matching the embedding dimension, and outputs hidden states of size 256.

3. **First Linear Transformation Layer:**

   The output passes through a fully connected (linear) layer that expands the feature space from 256 to 1024 units. This layer allows the model to learn a more complex representation of the sequential data.

4. **Dropout Layer:**

   The dropout layer with a rate of 0.5 follows the first linear transformation. This regularization technique helps prevent overfitting by randomly zeroing out a portion of the output features from the previous layer during training.

5. **Second Linear Transformation Layer:**

   The output from the dropout layer is then passed through another fully connected layer, which transfers the dimension from 1024 down to French vocabulary size, corresponding to the size of the target vocabulary (plus one for potential padding). This layer's output can be interpreted as unnormalized log probabilities for each token in the target vocabulary.

6. **Output Layer:**

   A log softmax activation function is applied along the second dimension of the output from the second linear transformation. This converts the linear outputs to logistic probabilities, making the model's output suitable for multi-class classification tasks. The log softmax output is particularly useful in conjunction with the Negative Log Likelihood (NLL) loss during training.

LSTM

RNN face challenges in retaining long-term dependencies and encounter the vanishing and exploding gradients problem, which Long Short-Term Memory (LSTM) networks address with their advanced architecture. LSTMs preserve information over extended sequences via a separate memory cell that is regulated by three gates: input gate, forget gate and output gate. In the implemented model, all the structures are similar to Simple RNN but substitute the RNN layer with the LSTM layer by calling nn.LSTM. LSTMs maintain important information over extended sequences from long period of time rather than RNN in short.

GRU

The Gated Recurrent Unit (GRU) was developed as a simplified alternative to the LSTM, eliminating the need for a separate memory cell state. Instead, GRUs use a "candidate activation vector" that is adjusted through two gates: the reset and update gates. In the model implement. All the structures are similar to LSTM but substitute the LSTM layer with the GRU layer by calling nn.GRU. GRU helps the model capture dependencies over time within the input sequence and reduces the training time compared with LSTM.

Seq2Seq LSTM

Seq2Seq is a many-to-many neural network architecture consisting of two key components: an encoder and a decoder, working in tandem to translate one sequence into another. This model's standout feature is its versatility, accommodating sequences of variable lengths for both inputs and outputs with ease.

Encoder layer

1. **Embedding layer:**

   The first embedding layer is the initial processing step that receives input sequences with English vocabulary size. It converts these input token indices into dense vectors of size 256. This transformation facilitates the learning of token representations that capture semantic meanings.

2. **LSTM layer:**

   Following the LSTM layer, the model condenses information into what is known as the hidden state, or cell state vectors. The encoder retains solely these internal states. The purpose of this context vector is to capture the essence of all input elements, thereby aiding the decoder in making precise predictions.

Decoder layer

1. **Embedding layer:**

   Input to the decoder will be the input sequences with French vocabulary size. The embedding layer converts token indices from French vocabularies into dense vector representations of a specified embedding dimension.

2. **LSTM layer:**

   LSTM layer uses this hidden state from the encoder to generate the output sequence step-by-step, taking the embedded target language sequences as additional input.

3. **Linear Transformation Layer and Dropout:**

   After decoding, the model applies a linear transformation to scale up features from the hidden dimension. A ReLU activation function follows to introduce non-linearity, making the model capable of learning more complex patterns. A dropout layer is then applied to prevent over-fitting by randomly zeroing out some of the features. After that, another linear transformation maps the 1024-dimensional features down to the size of the target vocabulary, preparing the output for probability distribution calculation over possible target tokens.

Seq2Seq GRU

In order to make training time shorter, GRU has been integrated into Seq2Seq. Therefore, all the structures are similar to Seq2Seq LSTM but substitute the LSTM layer with the GRU layer by calling nn. GRU in both the encoder and decoder.

Seq2Seq Transformer

The MarianMTModel[8] is employed and fine-tuned to suit our task for further analysis. It is designed and optimized for machine translation tasks based on the Transformer architecture. Transformer completely relies on the self-attention mechanism to process sequence data, abandoning the previous model that relied on RNNs.[9] The MarianMTModel's Transformer architecture also consists of an encoder and a decoder. Below are the crucial features to help understand the model:

1. **Encoder:**

   consists of multiple identical layers, each layer has two sub-layers. The first sub-layer is a multi-head self-attention mechanism, and the second sub-layer is a simple, positionally fully connected feed-forward network. There is a residual connection around each sub-layer, followed by layer normalization.

2. **Decoder:**

   Also consists of multiple identical layers, but each layer has three sub-layers. The decoder introduces a third sub-layer that processes the output of the encoder. The self-attention mechanism in the decoder is modified to prevent positions from looking backwards, ensuring predictions rely only on known outputs.

3. **Self-attention mechanism:**

   Allows the model to consider all positions in the sequence when processing the input sequence, dynamically assigning different weights to the output of each position. This is accomplished by calculating an attention score, which determines the importance of each word to the current word.

4. **Bullish attention:**

   The attention is divided into multiple "heads", with each head learning different aspects of the sequence. Doing so is efficient as different attention heads can capture different types of information.

5. **Position encoding:**

   Positional encoding is applied to handle the position information of words in the sequence, allowing the model to take advantage of the order of the words.

Numerical Experiments

Benchmarks Summary

The following tables demonstrate models' performance, evaluated by different metrics. All of the models look promising during training, and are having fluctuations in testing.

Fine-Tuning Experiments with Batch Size and Epochs

At the beginnig of the experiment, the The expected result is that the GRU seq2seq or LSTM seq2seq models will perform best because Seq2Seq specializes in sentence-to-sentence training. Conversely, the Vanilla RNN is anticipated to be the least effective model. Additionally, models related to GRU are expected to outperform those associated with LSTM, offering time-saving benefits. Moreover, models employing an encoder-decoder architecture should surpass those relying solely on basic, pure RNNs. Batch size and epoch duration are two critical variables for training the models and comparing their performance. The primary reason for selecting a batch size of 1024 is to mitigate the high time complexity associated with model training. Furthermore, a smaller number of epochs reduces the overall training time. A comparison of training losses in Table 1 clearly shows that the LSTM model incurs the highest training loss, while the Vanilla RNN exhibits the lowest. In contrast, when assessing the combination of test BLEU, METEOR scores, and cosine similarity, the GRU seq2seq model achieves the highest results, with the LSTM model showing the lowest performance in terms of batch size and epochs. The underperformance of the LSTM model can be attributed to an insufficient number of epochs for proper training. Therefore, increasing the epoch count is a promising direction for future experiments. According to Table 2 (batch size = 1024 and epoch = 50), the LSTM seq2seq emerges as the best model, contrary to expectations, while the standalone LSTM model performs the worst when comparing test scores (BLEU, METEOR, and cosine similarity). This outcome suggests that LSTM requires the seq2seq framework to leverage its long-term memory effectively. The final experiment, combining a batch size of 512 and 100 epochs, shows that the Seq2Seq GRU model excels in both BLEU scores and cosine similarity, with the Seq2Seq LSTM as a close second (achieving the highest METEOR score) but with the standalone LSTM performing worst. This confirms a new perspective based on the analysis of the previous tables. Overall, seq2seq models, whether LSTM or GRU, outperform simple RNNs. Meanwhile, across all tables, the pure GRU model consistently surpasses the pure LSTM model, so if the model without the GRU will be a better choice.

Model	Train Loss	Train BLEU Score	Test BLEU Score	Test METEOR Score	Cosine Similarity
Vanilla RNN	1.661	0.631	0.311	0.258	0.410
LSTM RNN	3.521	0.127	0.152	0.124	0.279
GRU RNN	2.825	0.195	0.206	0.203	0.394
Seq2Seq GRU	1.969	0.60	0.351	0.373	0.477
Seq2Seq LSTM	2.271	0.278	0.263	0.270	0.407

Table 1: The scores of batch size = 1024 and epoch = 20 on the train and test datasets

Model	Train Loss	Train BLEU Score	Test BLEU Score	Test METEOR Score	Cosine Similarity
Vanilla RNN	0.384	0.784	0.307	0.251	0.410
LSTM RNN	2.987	0.164	0.168	0.159	0.301
GRU RNN	1.389	0.573	0.352	0.299	0.468
Seq2Seq GRU	1.294	0.488	0.450	0.480	0.579
Seq2Seq LSTM	1.334	0.651	0.477	0.491	0.600

Table 2: The scores of batch size = 1024 and epoch = 50 on the train and test datasets

Model	Train Loss	Train BLEU Score	Test BLEU Score	Test METEOR Score	Cosine Similarity
Vanilla RNN	0.153	0.823	0.306	0.246	0.372
LSTM RNN	0.129	0.658	0.238	0.207	0.333
GRU RNN	0.175	0.744	0.311	0.230	0.365
Seq2Seq GRU	0.014	0.822	0.704	0.774	0.869
Seq2Seq LSTM	0.018	0.819	0.677	0.787	0.854

Table 3: The scores of batch size = 512 and epoch = 100 on the train and test datasets

Fine-Tuning Experiments with Hidden Size, Hidden Layers and Bidirectional

The Table 4 illustrates that bidirectional models outperform their unidirectional counterparts across all tested configurations, showing improved BLEU scores for both training and validation. This indicates that bidirectional architectures enhance translation accuracy by effectively utilizing contextual clues from both directions of the text. However, expanding the model from two to three layers leads to a marginal decrease in validation BLEU scores, hinting at potential overfitting or a plateau in performance gains from additional complexity. For single-layer models, increasing the hidden unit size from 256 to 512 yields a modest enhancement in performance, suggesting that larger models have a better capacity for capturing information, though the benefits of this increment are not pronounced. The consistently low training losses suggest that all configurations are achieving a good fit on the training data. Yet, the discrepancy between high training and lower validation BLEU scores point to insufficient data, overfitting, and the necessity for further regularization or data augmentation to bolster the model's generalization capabilities. It is also quite noticeable as model become increasingly complicated, the training time gets longer and for models set to be bidirectional the train time almost doubles.

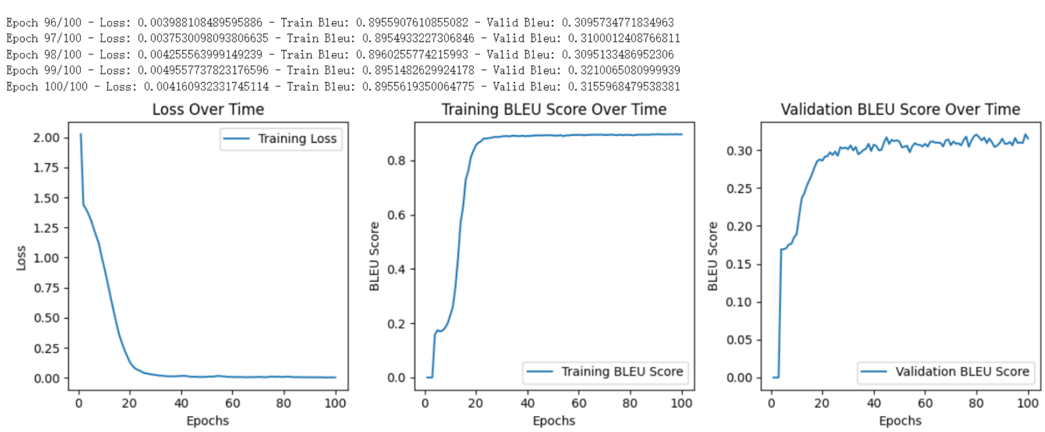


Figure 3: Sample output graphs for 2 hidden layers 512 hidden units bidirectional LSTM. For the complete experiment trials, refer to the "Test Space for LSTM RNN" in supplementary file.

Model: LSTM Epochs: 100	Train Loss	Train BLEU Score	Valid BLEU Score
num_layers: 1 hidden size: 256 unidirectional	0.029	0.871	0.313
num_layers: 1 hidden size: 256 bidirectional	0.005	0.895	0.314
num_layers: 1 hidden size: 512 bidirectional	0.007	0.896	0.320
num_layers: 2 hidden size: 512 bidirectional	0.004	0.896	0.316
num_layers: 3 hidden size: 512 bidirectional	0.006	0.895	0.303

Table 4: Fine-tuning experiments for LSTM model (batch_size = 64).

Fine-Tuning Seq2Seq Transformer

In Figure 4 we observe the performance of the Seq2Seq Transformer model, which shows high accuracy on smaller datasets and can predict sentences with previously seen words very effectively . However, this acuity falters when transitioning to larger data sets. This further verifies our idea that our model is built correctly and can achieve extremely high accuracy during training. However, at test time, our dataset introduced data quality issues, such as too many prepositions or too few occurrences of some nouns, and the model disproportionately learned common phrases at the expense of less common phrases, thereby messing with the model's predictions. This directly results in our model performing well in familiar environments but not performing well when facing a wider range of language environments. We need to further fully consider training strategies that ensure that common and rare expressions are represented and learned, which shall be discussed in detail in the following section.

```
print(((predicted_french_sentence)))
```

```
that prejudice grew into a dogma
ce préjugé est devenu un dogme
[[  8 395 396 113   3 397   0   0   0   0   0   0   0   0   0   0   0   0
    0   0]]
[[ 24 404   7 405   6 406   0   0   0   0   0   0   0   0   0   0   0   0
    0   0]]
ce préjugé est devenu un dogme
```

```
for sentence_en, sentence_fr in zip(test_sentences_en, predicted_sentences_fr):
        print("English Sentence:", sentence_en)
        print("Predicted French Translation:", sentence_fr)
```

```
English Sentence: This is a test sentence.
Predicted French Translation: C'est une phrase d'essai.
English Sentence: Another test sentence.
Predicted French Translation: Une autre phrase d'essai.
```

Figure 4: Sample outstanding output of Seq2Seq Transformer

Discussion and Improvements

Overall, our models consistently exhibit low training losses that diminish and plateau as training progresses, with high training BLEU scores, proving the robustness of our model architecture. However, a persisting problem is that BLEU scores during validation and testing almost always stay the same, indicating a shortfall in generalization. Although we have conducted a lot of hyperparameter tuning experiments, where the train loss and train BLEU had significant improvement, (i.e. converge more rapidly, lower loss and higher score), this phenomenon has not been well resolved.

We infer that the primary causes of this issue are insufficient data and inadequate data processing. Our English and French vocabulary sizes encompass close to 5,000 words each, with prepositions forming a substantial proportion, and our corpus has roughly 2,600 sample sentences. Furthermore, since we obtain subtitles from different YouTube channels, such as those focused on politics, military, or nature, the word frequency distribution is highly skewed. For instance, specific terms like "Amazon rainforest" might only appear once within a documentary, thus being scarcely represented in our dataset. This scarcity impairs the model's ability to comprehend such terms during evaluation.

The solution is also straightforward: in future tests, we should augment the dataset with a wider range of bilingual subtitles, especially those covering previously unrepresented topics and vocabulary. Improve the handling of rare and unknown words using subword tokenization methods such as Byte Pair Encoding (BPE) to enhance the model's ability to handle new terms.

Further enhancements include enriching the language representation with pre-trained embeddings such as fastText or BERT. Perhaps one should also consider incorporating as-yet-unexploited regularization methods, including weight decay (L2 regularization) or label smoothing, to prevent overfitting. Advanced strategies usually require more detailed hyperparameter optimization to discover better model configurations, which extends to the use of ensemble methods to integrate multiple translation models to improve translation quality. In addition, data augmentation strategies such as back-translation and paraphrase, coupled with beam search during inference, can also help us achieve a good balance between the quality and speed of model training.

Conclusion

In summary, our exploration of the field of automatic language translation through YouTube video scripts provides a promising approach to improving machine translation capabilities. We have established a foundational process and platform, starting from acquiring bilingual datasets, to fine-tuning sequence-to-sequence models through an innovative integration of word2vec, RNN and advanced Transformer architecture.

Despite challenges in model generalization (as evidenced by the stable level of validation BLEU scores), our approach shows great potential in capturing language nuances and effectively translating them. Our iterative approach to data, implementation of subword tokenization, and use of pretrained embeddings lay the foundation for a model capable of adapting to the rich variability of human language. Our work confirms the importance of datasets in the training process and the need for models that can be extended by including category-specific captions.

Future efforts will focus on expanding our dataset, refining our preprocessing strategies, and exploring the integration of cutting-edge models and techniques. For future investigations, we could consider directions for improvements of the model include model merging: SWA,[10] EMA,[11] accelerated inference: Mixed Precision (bf16), 8-bit/1-bit qunatlized,[12] Adversarial Attacks: AWP,[13] and better pre-train models: DeBERTa v3.[14] The insights gained from this project not only contribute to academic discussions but also have significant real-world applications.

References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[2] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in Neural Information Processing Systems, 2014, pp. 3104-3112. [Online]. Available: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

[3] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, 2017, pp. 5998-6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[4] Projjal1, "English-French-Translator-RNN," GitHub repository, 2020. [Online]. Available: https://github.com/projjal1/English-French-Translator-RNN

[5] Z. Thoutt, "Language-translation-rnn," GitHub repository, 2017. [Online]. Available: https://github.com/zackthoutt/language-translation-rnn

[6] Google Colab. [Online]. Available: https://colab.research.google.com/.

[7] NVIDIA Corporation, "NVIDIA A100 Tensor Core GPU Architecture," [Online]. Available: https://www.nvidia.com/en-us/data-center/a100/.

[8] Marian-NMT, "Marian: Fast Neural Machine Translation in C++," GitHub repository, [Online]. Available: https://github.com/marian-nmt/marian, Accessed on: Apr. 4, 2024.

[9] A. Vaswani et al., "Attention is all you need," Advances in Neural Information Processing Systems, vol. 30, 2017. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[10] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, “Averaging weights leads to wider optima and better generalization,” Mar. 2018.

[11] F. Klinker, “Exponential moving average versus moving exponential average,” Math. Semesterber., vol. 58, no. 1, pp. 97–107, Dec. 2020.

[12] T. Piao, I. Cho, and U. Kang, “Sensimix: Sensitivity-aware 8-bit index 1-bit value mixed precision quantization for bert compression,” PLOS ONE, vol. 17, no. 4, Art. no. e0265621, Apr. 2022.

[13] N. Akhtar, A. Mian, N. Kardan, and M. Shah, “Advances in adversarial attacks and defenses in computer vision: A survey,” Aug. 2021.

[14] P. He, X. Liu, J. Gao, and W. Chen, “Deberta: Decoding-enhanced bert with disentangled attention,” Jun. 2020.

Appendix

This section provides helpful supplementary materials involved in our project.

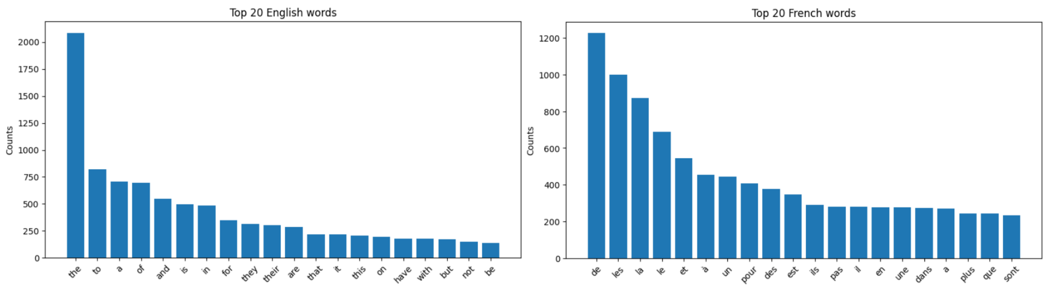


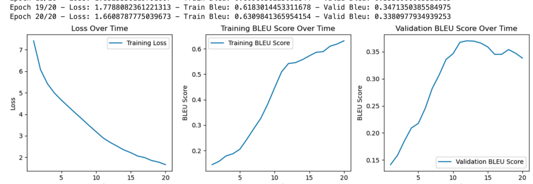
Figure 5: Top 20 most frequent vocabularies for English and French datasets.

```
vector = english_model.wv['their']
similarity = english_model.wv.similarity('their', 'end')
similar_words = english_model.wv.most_similar('their', topn=5)
vector, similarity, similar_words
```

```
(array([ 0.02631398,  0.32628238,  0.12123577,  0.00956412, -0.01134941,
        -0.24409564,  0.30601507,  0.5179731 , -0.29293478, -0.19796938,
        -0.06882792, -0.27393964, -0.00645052,  0.05799846,  0.14829278,
        -0.15073557,  0.09058712, -0.20003031, -0.12899874, -0.48126695,
         0.16893782,  0.21450734,  0.3314231 , -0.16032909,  0.00868978,
         0.08062547, -0.02602514, -0.04836071, -0.21493301, -0.0025136 ,
         0.17421907, -0.11512037,  0.40916777, -0.19147703,  0.01454104,
         0.29800364,  0.11956123, -0.10813519, -0.28113493, -0.4337156 ,
         0.11604215, -0.18278779, -0.16004021, -0.03567465,  0.18267275,
        -0.01646074, -0.12224813, -0.10065123,  0.10576606,  0.06070855,
         0.04286717, -0.22364688, -0.14575155,  0.058098  , -0.13144866,
         0.2052925 ,  0.18498965, -0.15667978, -0.02693537,  0.02990344,
         0.12304795, -0.00322075,  0.1320696 ,  0.03590341, -0.3086271 ,
         0.31994396,  0.01920592,  0.32331815, -0.38248232,  0.07887494,
        -0.0409305 ,  0.29269794,  0.33187252,  0.07246795,  0.15374538,
        -0.06395937,  0.13671221,  0.00347286, -0.28099835, -0.01362745,
        -0.20478383,  0.03597464, -0.17215738,  0.19238773,  0.0292585 ,
        -0.09046405,  0.04263978,  0.10174638,  0.17937216,  0.16281635,
         0.22757329,  0.0752079 ,  0.08272428, -0.05327092,  0.3171552 ,
         0.20937495, -0.06356742, -0.17636022, -0.1288417 ,  0.08896936],
       dtype=float32),
 0.98828334,
 [('males', 0.9946478605270386),
  ('front', 0.9939694404602051),
  ('themselves', 0.9937782883644104),
  ('border', 0.993556022644043),
  ('food', 0.9935117959976196)])
```

Figure 6: Sample output of word2vec vector for "their" with its most frequent occur words.

```
/var/folders/p6/vtpc4hm50lb015mtg41x194m0000gn/T/ipykernel_58224/3618713725.py:5: UserWarning: 'has_mps' is deprecated, ple
ase use 'torch.backends.mps.is_built()'
  elif torch.has_mps:
Epoch 1/20 - Loss: 7.415560245513916 - Train Bleu: 0.1447356253326741 - Valid Bleu: 0.14141537716869718
Epoch 2/20 - Loss: 6.072548230489095 - Train Bleu: 0.15747513083916456 - Valid Bleu: 0.15860180748362807
Epoch 3/20 - Loss: 5.420052210489909 - Train Bleu: 0.17891386035701903 - Valid Bleu: 0.18525773312446167
Epoch 4/20 - Loss: 4.99006716410319 - Train Bleu: 0.18742995795986372 - Valid Bleu: 0.20910202797733746
Epoch 5/20 - Loss: 4.661862373352051 - Train Bleu: 0.2052200677642042 - Valid Bleu: 0.21750872604705296
Epoch 6/20 - Loss: 4.360051314036052 - Train Bleu: 0.24391702581085956 - Valid Bleu: 0.24549863891155416
Epoch 7/20 - Loss: 4.064098437627156 - Train Bleu: 0.28563007553310343 - Valid Bleu: 0.2820375846616255
Epoch 8/20 - Loss: 3.7735440731048584 - Train Bleu: 0.3257716010341807 - Valid Bleu: 0.30696875536299895
Epoch 9/20 - Loss: 3.475375175476074 - Train Bleu: 0.3814824096446351 - Valid Bleu: 0.3357138305512089
Epoch 10/20 - Loss: 3.1927126248677573 - Train Bleu: 0.44577364134468433 - Valid Bleu: 0.3462246842986088
Epoch 11/20 - Loss: 2.907064358393351 - Train Bleu: 0.508577276999537 - Valid Bleu: 0.3673655746784887
Epoch 12/20 - Loss: 2.699631134668986 - Train Bleu: 0.5417662989762774 - Valid Bleu: 0.37003226018449314
Epoch 13/20 - Loss: 2.524954319000244 - Train Bleu: 0.5451970086382384 - Valid Bleu: 0.36943172786919626
Epoch 14/20 - Loss: 2.346240758895874 - Train Bleu: 0.5568535699371936 - Valid Bleu: 0.365848443686918
Epoch 15/20 - Loss: 2.2172998587290444 - Train Bleu: 0.5717652538360904 - Valid Bleu: 0.3587748573955468
Epoch 16/20 - Loss: 2.0607371727625527 - Train Bleu: 0.5859436055108803 - Valid Bleu: 0.34506304781468444
Epoch 17/20 - Loss: 1.9891733725865681 - Train Bleu: 0.5884969751280203 - Valid Bleu: 0.3450506807592455
Epoch 18/20 - Loss: 1.8523356119791667 - Train Bleu: 0.6098056225082344 - Valid Bleu: 0.3538900063521983
Epoch 19/20 - Loss: 1.7788082361221313 - Train Bleu: 0.6183014453311678 - Valid Bleu: 0.3471350385584975
Epoch 20/20 - Loss: 1.6608787775039673 - Train Bleu: 0.6309841365954154 - Valid Bleu: 0.3380977934939253
```



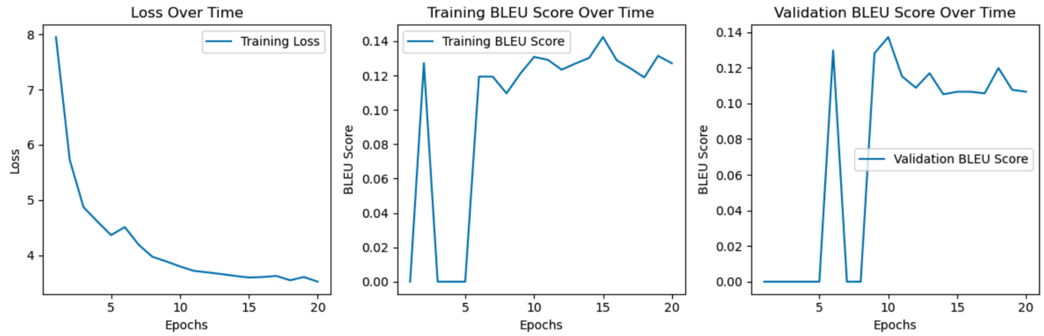

```
Training complete.
```

```
basic_rnn_model.eval()
prediction_with_translation_scores(basic_rnn_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
model_similarity_score(basic_rnn_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
```

```
Average BLEU Score: 0.3113
Average METEOR Score: 0.2578
Average Similarity Score 0.4104325401
```

Figure 7: Basic batch size = 1024 and epochs = 20

```
Epoch 1/20 - Loss: 7.951651732126872 - Train Bleu: 0 - Valid Bleu: 0
Epoch 2/20 - Loss: 5.726847807566325 - Train Bleu: 0.1272829724159886 - Valid Bleu: 0
Epoch 3/20 - Loss: 4.871510028839111 - Train Bleu: 0 - Valid Bleu: 0
Epoch 4/20 - Loss: 4.611642837524414 - Train Bleu: 0 - Valid Bleu: 0
Epoch 5/20 - Loss: 4.365040461222331 - Train Bleu: 0 - Valid Bleu: 0
Epoch 6/20 - Loss: 4.510002454121907 - Train Bleu: 0.11941967889516485 - Valid Bleu: 0.1297619575634719
Epoch 7/20 - Loss: 4.190679550170898 - Train Bleu: 0.11935560114653061 - Valid Bleu: 0
Epoch 8/20 - Loss: 3.972353935241699 - Train Bleu: 0.10964920390406709 - Valid Bleu: 0
Epoch 9/20 - Loss: 3.888540267944336 - Train Bleu: 0.12114568470426407 - Valid Bleu: 0.12811480155945082
Epoch 10/20 - Loss: 3.797919670740763 - Train Bleu: 0.13085337894958043 - Valid Bleu: 0.13724012045784528
Epoch 11/20 - Loss: 3.7174507776896157 - Train Bleu: 0.1291694644852545 - Valid Bleu: 0.1152140750129739
Epoch 12/20 - Loss: 3.689483642578125 - Train Bleu: 0.12341609677411225 - Valid Bleu: 0.10879313523801992
Epoch 13/20 - Loss: 3.6592883268992105 - Train Bleu: 0.12701334209763748 - Valid Bleu: 0.1169486841179445
Epoch 14/20 - Loss: 3.6253933111826577 - Train Bleu: 0.1303054049136067 - Valid Bleu: 0.1051202842109767
Epoch 15/20 - Loss: 3.596601406733195 - Train Bleu: 0.14242358023119558 - Valid Bleu: 0.10656000432042387
Epoch 16/20 - Loss: 3.605581045150757 - Train Bleu: 0.12891800752891053 - Valid Bleu: 0.10656000432042387
Epoch 17/20 - Loss: 3.6251488526662192 - Train Bleu: 0.12418942733095353 - Valid Bleu: 0.10564697893835612
Epoch 18/20 - Loss: 3.5466222763061523 - Train Bleu: 0.11889547402065137 - Valid Bleu: 0.11987318497734485
Epoch 19/20 - Loss: 3.605401357014974 - Train Bleu: 0.131484486849913 - Valid Bleu: 0.10764015380207402
Epoch 20/20 - Loss: 3.5212451616923013 - Train Bleu: 0.12715410875272432 - Valid Bleu: 0.10656000432042387
```



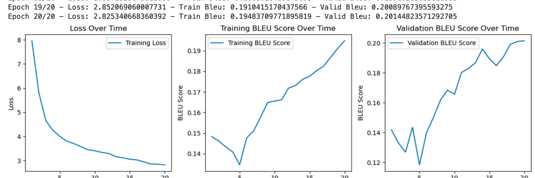

```
Training complete.
```

```
lstm_rnn_model.eval()
prediction_with_translation_scores(lstm_rnn_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
model_similarity_score(lstm_rnn_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
```

```
Average BLEU Score: 0.1517
Average METEOR Score: 0.1238
Average Similarity Score 0.2787868720
```

Figure 8: LSTM batch size = 1024 and epochs = 20

```
Epoch 1/20 - Loss: 7.960601647694905 - Train Bleu: 0.1483041063234454 - Valid Bleu: 0.14171967564618304
Epoch 2/20 - Loss: 5.805113792419434 - Train Bleu: 0.14618641380500766 - Valid Bleu: 0.13292845915853088
Epoch 3/20 - Loss: 4.652523676554362 - Train Bleu: 0.14332297590350923 - Valid Bleu: 0.12686560843323058
Epoch 4/20 - Loss: 4.24797264734904 - Train Bleu: 0.14089449127598072 - Valid Bleu: 0.14351934414174827
Epoch 5/20 - Loss: 4.006799856821696 - Train Bleu: 0.13459587287487806 - Valid Bleu: 0.11837755991084298
Epoch 6/20 - Loss: 3.8004908561706543 - Train Bleu: 0.14766329246749924 - Valid Bleu: 0.13979273535335002
Epoch 7/20 - Loss: 3.704313357671102 - Train Bleu: 0.15108150218973632 - Valid Bleu: 0.150159068532076
Epoch 8/20 - Loss: 3.581387519836426 - Train Bleu: 0.15783244111842806 - Valid Bleu: 0.16167859071313206
Epoch 9/20 - Loss: 3.449831962585449 - Train Bleu: 0.16485019598097486 - Valid Bleu: 0.16839495089278428
Epoch 10/20 - Loss: 3.406610091527303 - Train Bleu: 0.165558180401521 - Valid Bleu: 0.16556316416515635
Epoch 11/20 - Loss: 3.3392818768819175 - Train Bleu: 0.16615250055958578 - Valid Bleu: 0.18012344087238694
Epoch 12/20 - Loss: 3.2930779457092285 - Train Bleu: 0.17187359772376723 - Valid Bleu: 0.18293224197415864
Epoch 13/20 - Loss: 3.164853890736898 - Train Bleu: 0.1731075584752303 - Valid Bleu: 0.18665045945199993
Epoch 14/20 - Loss: 3.116924206415812 - Train Bleu: 0.17611776232660267 - Valid Bleu: 0.19592703936000158
Epoch 15/20 - Loss: 3.0558012326558432 - Train Bleu: 0.17754383221516765 - Valid Bleu: 0.18939187980773778
Epoch 16/20 - Loss: 3.030225912729899 - Train Bleu: 0.1803311273793665 - Valid Bleu: 0.18473242841688386
Epoch 17/20 - Loss: 2.9487077395121255 - Train Bleu: 0.182476468076099 - Valid Bleu: 0.19065035573412303
Epoch 18/20 - Loss: 2.8572359879811606 - Train Bleu: 0.1868028420145093 - Valid Bleu: 0.19910512832611693
Epoch 19/20 - Loss: 2.852069060007731 - Train Bleu: 0.1910415170437566 - Valid Bleu: 0.20089767395593275
Epoch 20/20 - Loss: 2.825340668360392 - Train Bleu: 0.19483709771895819 - Valid Bleu: 0.20144823571292705
```



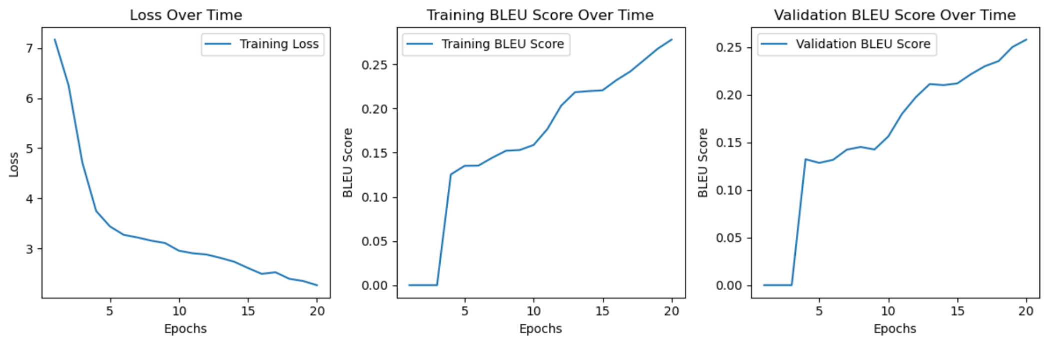

```
Training complete.
```

```
gru_rnn_model.eval()
prediction_with_translation_scores(gru_rnn_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
model_similarity_score(gru_rnn_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
```

```
Average BLEU Score: 0.2057
Average METEOR Score: 0.2027
Average Similarity Score 0.3943199391
```

Figure 9: GRU batch size = 1024 and epochs = 20

```
Epoch 1/20 - Loss: 7.164259592692058 - Train Bleu: 0 - Valid Bleu: 0
Epoch 2/20 - Loss: 6.252976417541504 - Train Bleu: 0 - Valid Bleu: 0
Epoch 3/20 - Loss: 4.712393283843994 - Train Bleu: 0 - Valid Bleu: 0
Epoch 4/20 - Loss: 3.74792488416036 - Train Bleu: 0.12529547553677162 - Valid Bleu: 0.13216936438293445
Epoch 5/20 - Loss: 3.4411110083262124 - Train Bleu: 0.13504962079021193 - Valid Bleu: 0.1283694831870541
Epoch 6/20 - Loss: 3.273523489634196 - Train Bleu: 0.13528735470550476 - Valid Bleu: 0.1315912871024569
Epoch 7/20 - Loss: 3.2207518418629966 - Train Bleu: 0.14421021294910985 - Valid Bleu: 0.14227345001287361
Epoch 8/20 - Loss: 3.157448132832845 - Train Bleu: 0.15211497807052424 - Valid Bleu: 0.14507569103986517
Epoch 9/20 - Loss: 3.1098477840423584 - Train Bleu: 0.15295850280511517 - Valid Bleu: 0.14242431156926824
Epoch 10/20 - Loss: 2.957255760828654 - Train Bleu: 0.15864535262602047 - Valid Bleu: 0.15613930713866978
Epoch 11/20 - Loss: 2.906479756037394 - Train Bleu: 0.17669778416020496 - Valid Bleu: 0.18000153958087317
Epoch 12/20 - Loss: 2.8816173871358237 - Train Bleu: 0.20314783137132236 - Valid Bleu: 0.19745299841664152
Epoch 13/20 - Loss: 2.8145905335744223 - Train Bleu: 0.2183273757624878 - Valid Bleu: 0.21110137066112375
Epoch 14/20 - Loss: 2.7385127544403076 - Train Bleu: 0.21959956531049038 - Valid Bleu: 0.20999795160161336
Epoch 15/20 - Loss: 2.613590399424235 - Train Bleu: 0.2205093800755803 - Valid Bleu: 0.21178277090362113
Epoch 16/20 - Loss: 2.49593456586202 - Train Bleu: 0.2320077322073398 - Valid Bleu: 0.22153079228778733
Epoch 17/20 - Loss: 2.5279725392659507 - Train Bleu: 0.24172809698305878 - Valid Bleu: 0.22986551882297654
Epoch 18/20 - Loss: 2.3978284994761148 - Train Bleu: 0.2546405210231055 - Valid Bleu: 0.23525990522848747
Epoch 19/20 - Loss: 2.3533500830332437 - Train Bleu: 0.26760965712980955 - Valid Bleu: 0.2500313697325329
Epoch 20/20 - Loss: 2.270758310953776 - Train Bleu: 0.27786131959673466 - Valid Bleu: 0.25775062714028396
```



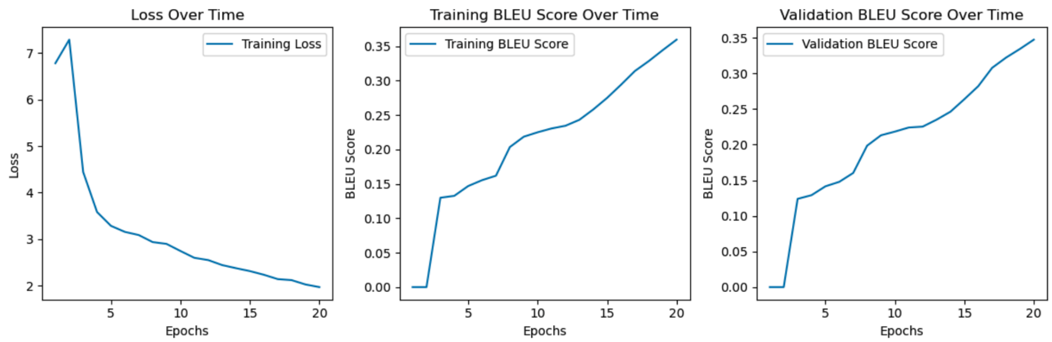

```
Training complete.
```

```
lstm_seq2seq_model.eval()
prediction_with_translation_scores_seq2(lstm_seq2seq_model, test_sentences, test_fr_sentences, english_tokens, french_token:
model_similarity_score_seq2(lstm_seq2seq_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
```

```
Average BLEU Score: 0.2632
Average METEOR Score: 0.2698
Average Similarity Score 0.4071105086
```

Figure 10: LSTM Seq2Seq batch size = 1024 and epochs = 20

```
Epoch 1/20 - Loss: 6.779940764109294 - Train Bleu: 0 - Valid Bleu: 0
Epoch 2/20 - Loss: 7.2875823974609375 - Train Bleu: 0 - Valid Bleu: 0
Epoch 3/20 - Loss: 4.444279193878174 - Train Bleu: 0.1298801728873776 - Valid Bleu: 0.12386981302573717
Epoch 4/20 - Loss: 3.5825560092926025 - Train Bleu: 0.13263925006427865 - Valid Bleu: 0.1290454585551937
Epoch 5/20 - Loss: 3.2856086095174155 - Train Bleu: 0.14687334135920896 - Valid Bleu: 0.14146202510050263
Epoch 6/20 - Loss: 3.156081438064575 - Train Bleu: 0.15542520337635649 - Valid Bleu: 0.14778919750917352
Epoch 7/20 - Loss: 3.0880399545033774 - Train Bleu: 0.16192316784696859 - Valid Bleu: 0.16022266370040913
Epoch 8/20 - Loss: 2.9369966983795166 - Train Bleu: 0.20355263615034386 - Valid Bleu: 0.19851151773001538
Epoch 9/20 - Loss: 2.8976932366689048 - Train Bleu: 0.2185837837848466 - Valid Bleu: 0.21311409649300575
Epoch 10/20 - Loss: 2.745437224706014 - Train Bleu: 0.2250474001222758 - Valid Bleu: 0.21828387758027593
Epoch 11/20 - Loss: 2.598745028177897 - Train Bleu: 0.23057890071812115 - Valid Bleu: 0.22404876548394323
Epoch 12/20 - Loss: 2.549862861633301 - Train Bleu: 0.23451788326488132 - Valid Bleu: 0.22527038584906073
Epoch 13/20 - Loss: 2.442719300587972 - Train Bleu: 0.24312842779970825 - Valid Bleu: 0.23502782597439228
Epoch 14/20 - Loss: 2.375560204188029 - Train Bleu: 0.25817087936807215 - Valid Bleu: 0.24623416912171583
Epoch 15/20 - Loss: 2.3129116694132485 - Train Bleu: 0.27503698051564424 - Valid Bleu: 0.2637842263928934
Epoch 16/20 - Loss: 2.234735647837321 - Train Bleu: 0.29414516173202754 - Valid Bleu: 0.2820543639125829
Epoch 17/20 - Loss: 2.14115842183431 - Train Bleu: 0.31411253479869405 - Valid Bleu: 0.307818066544608
Epoch 18/20 - Loss: 2.1186575094858804 - Train Bleu: 0.3285557128418408 - Valid Bleu: 0.3221632775110322
Epoch 19/20 - Loss: 2.025814175605774 - Train Bleu: 0.3444152540191416 - Valid Bleu: 0.33434531111007826
Epoch 20/20 - Loss: 1.9690689245859783 - Train Bleu: 0.3596859320703031 - Valid Bleu: 0.3473727728365737
```



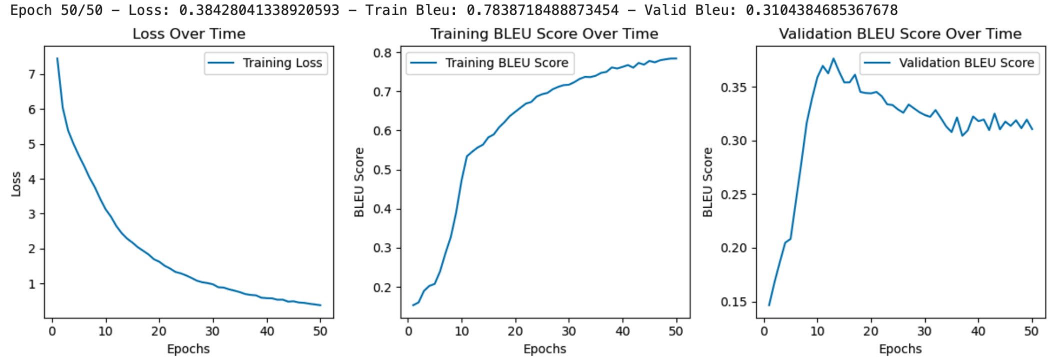

```
Training complete.
```

```
gru_seq2seq_model.eval()
prediction_with_translation_scores_seq2(gru_seq2seq_model, test_sentences, test_fr_sentences, english_tokens, french_tokens
model_similarity_score_seq2(gru_seq2seq_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
```

```
Average BLEU Score: 0.3511
Average METEOR Score: 0.3725
Average Similarity Score 0.4774767457
```

Figure 11: GRU Seq2Seq batch size = 1024 and epochs = 20

```
Epoch 1/50 - Loss: 7.437252521514893 - Train Bleu: 0.15290251905215121 - Valid Bleu: 0.14650581026155982
Epoch 2/50 - Loss: 6.03633975982666 - Train Bleu: 0.16004236539921105 - Valid Bleu: 0.16783641117849907
Epoch 3/50 - Loss: 5.382530530293782 - Train Bleu: 0.18938535123461833 - Valid Bleu: 0.1866712121167555
Epoch 4/50 - Loss: 5.002683003743489 - Train Bleu: 0.20234590201477118 - Valid Bleu: 0.20474353453712496
Epoch 5/50 - Loss: 4.666648070017497 - Train Bleu: 0.20741199131639512 - Valid Bleu: 0.20820922790555835
Epoch 6/50 - Loss: 4.361829439798991 - Train Bleu: 0.23931340281823296 - Valid Bleu: 0.24376383561605316
Epoch 7/50 - Loss: 4.031380891799927 - Train Bleu: 0.28564941108006947 - Valid Bleu: 0.27892931949714656
Epoch 8/50 - Loss: 3.7534772555033364 - Train Bleu: 0.32716989331230883 - Valid Bleu: 0.3162165591417702
Epoch 9/50 - Loss: 3.420487642288208 - Train Bleu: 0.39002814136747577 - Valid Bleu: 0.3391637609805455
Epoch 10/50 - Loss: 3.1260035037994385 - Train Bleu: 0.471550484513033 - Valid Bleu: 0.3587588679324713
Epoch 11/50 - Loss: 2.9134204387664795 - Train Bleu: 0.5337811616217514 - Valid Bleu: 0.3693419626825327
Epoch 12/50 - Loss: 2.645917018254598 - Train Bleu: 0.5455850906442117 - Valid Bleu: 0.3623543995563805
Epoch 13/50 - Loss: 2.4430274168650308 - Train Bleu: 0.5561807807389163 - Valid Bleu: 0.37620061140469696
Epoch 14/50 - Loss: 2.288489023844401 - Train Bleu: 0.5634671981236622 - Valid Bleu: 0.3644577574963587
Epoch 15/50 - Loss: 2.173710664113363 - Train Bleu: 0.5822306418777086 - Valid Bleu: 0.3538961974034414
Epoch 16/50 - Loss: 2.04194704691569 - Train Bleu: 0.5895534959515374 - Valid Bleu: 0.3539790418392403
Epoch 17/50 - Loss: 1.9396167993545532 - Train Bleu: 0.6077680257785526 - Valid Bleu: 0.3611514024794326
Epoch 18/50 - Loss: 1.8348151048024495 - Train Bleu: 0.6211322661261066 - Valid Bleu: 0.34504419938526526
Epoch 19/50 - Loss: 1.6977985699971516 - Train Bleu: 0.6366749996408982 - Valid Bleu: 0.34404526557601806
Epoch 20/50 - Loss: 1.6273617347081502 - Train Bleu: 0.6474227033568594 - Valid Bleu: 0.3437557799207172
Epoch 21/50 - Loss: 1.5119163990020752 - Train Bleu: 0.6580863564053829 - Valid Bleu: 0.34505780227572147
Epoch 22/50 - Loss: 1.4333912134170532 - Train Bleu: 0.668551379446683 - Valid Bleu: 0.3409306623279685
Epoch 23/50 - Loss: 1.3354140520095825 - Train Bleu: 0.6727220288528634 - Valid Bleu: 0.3335169861446071
Epoch 24/50 - Loss: 1.2952004671096802 - Train Bleu: 0.6865708100396213 - Valid Bleu: 0.33285729173530015
Epoch 25/50 - Loss: 1.2354603211085002 - Train Bleu: 0.6924590919157296 - Valid Bleu: 0.3287758836714302
Epoch 26/50 - Loss: 1.1643932660420735 - Train Bleu: 0.695778681687405 - Valid Bleu: 0.3257067391943956
Epoch 27/50 - Loss: 1.0853718916575115 - Train Bleu: 0.7052304953018158 - Valid Bleu: 0.33333772785393356
Epoch 28/50 - Loss: 1.0404391884803772 - Train Bleu: 0.7112053083020181 - Valid Bleu: 0.3297385643102647
Epoch 29/50 - Loss: 1.0163801709810893 - Train Bleu: 0.7156674466092718 - Valid Bleu: 0.32618526834899925
Epoch 30/50 - Loss: 0.9816747903823853 - Train Bleu: 0.7167146990026159 - Valid Bleu: 0.32360103091825426
Epoch 31/50 - Loss: 0.8976060748100281 - Train Bleu: 0.7231608808575897 - Valid Bleu: 0.3218644146043182
Epoch 32/50 - Loss: 0.8875878651936849 - Train Bleu: 0.7316373942163007 - Valid Bleu: 0.3281992220476467
Epoch 33/50 - Loss: 0.8376768032709757 - Train Bleu: 0.7367953479990954 - Valid Bleu: 0.32068783474204077
Epoch 34/50 - Loss: 0.8002957304318746 - Train Bleu: 0.7361867115144765 - Valid Bleu: 0.31284341844889924
Epoch 35/50 - Loss: 0.7587874531745911 - Train Bleu: 0.7397545065792159 - Valid Bleu: 0.3076563496794106
Epoch 36/50 - Loss: 0.7036462823549906 - Train Bleu: 0.7470380691073977 - Valid Bleu: 0.3212973516733347
Epoch 37/50 - Loss: 0.6797142028808594 - Train Bleu: 0.7495979737239051 - Valid Bleu: 0.30419838067873545
Epoch 38/50 - Loss: 0.6667593717575073 - Train Bleu: 0.7610177605494627 - Valid Bleu: 0.30927545333074263
Epoch 39/50 - Loss: 0.5995073119799296 - Train Bleu: 0.7582259996752139 - Valid Bleu: 0.32225309240957
Epoch 40/50 - Loss: 0.5854669411977133 - Train Bleu: 0.7623726735581298 - Valid Bleu: 0.31780693493957046
Epoch 41/50 - Loss: 0.5810315211613973 - Train Bleu: 0.7670186716813443 - Valid Bleu: 0.31935528796026225
Epoch 42/50 - Loss: 0.541862428188324 - Train Bleu: 0.7603661229735417 - Valid Bleu: 0.3095176235460431
Epoch 43/50 - Loss: 0.5439172585805258 - Train Bleu: 0.7725541807629557 - Valid Bleu: 0.3249092708187709
Epoch 44/50 - Loss: 0.4869185785452525 - Train Bleu: 0.7681355658306932 - Valid Bleu: 0.3102040956388044
Epoch 45/50 - Loss: 0.49689318736394245 - Train Bleu: 0.7776657018286718 - Valid Bleu: 0.31742058674482737
Epoch 46/50 - Loss: 0.46212974190711975 - Train Bleu: 0.7740818082307184 - Valid Bleu: 0.3134765678949109
Epoch 47/50 - Loss: 0.4517318805058797 - Train Bleu: 0.7792853553348188 - Valid Bleu: 0.3186324358118328
Epoch 48/50 - Loss: 0.4247293968995412 - Train Bleu: 0.7818490438719474 - Valid Bleu: 0.31126660800198463
Epoch 49/50 - Loss: 0.4066040813922882 - Train Bleu: 0.7837194642093142 - Valid Bleu: 0.3192680931453272
Epoch 50/50 - Loss: 0.38428041338920593 - Train Bleu: 0.7838718488873454 - Valid Bleu: 0.3104384685367678
```



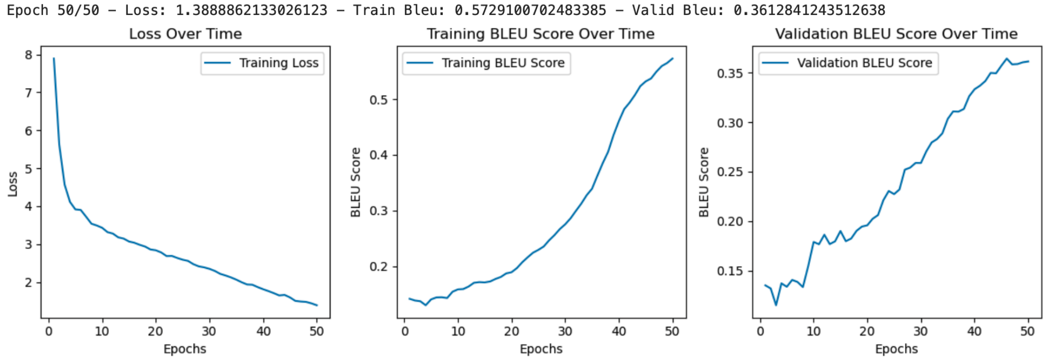

```
Training complete.
```

```
basic_rnn_model.eval()
prediction_with_translation_scores(basic_rnn_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
model_similarity_score(basic_rnn_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
```

```
Average BLEU Score: 0.3065
Average METEOR Score: 0.2511
Average Similarity Score 0.4096588892
```

Figure 12: Basic batch size = 1024 and epochs = 50

```
Epoch 1/50 - Loss: 7.89097785949707 - Train Bleu: 0.14134259853883008 - Valid Bleu: 0.1351003369574105
Epoch 2/50 - Loss: 5.624489625295003 - Train Bleu: 0.13824384672480425 - Valid Bleu: 0.13197425220448666
Epoch 3/50 - Loss: 4.568843523661296 - Train Bleu: 0.13692147720721162 - Valid Bleu: 0.1149976691751243
Epoch 4/50 - Loss: 4.112732887268066 - Train Bleu: 0.12975384111584748 - Valid Bleu: 0.13715234251722863
Epoch 5/50 - Loss: 3.9154974619547525 - Train Bleu: 0.14023509423248792 - Valid Bleu: 0.13364376274259307
Epoch 6/50 - Loss: 3.9015142917633057 - Train Bleu: 0.14384960947438163 - Valid Bleu: 0.1406113645375631
Epoch 7/50 - Loss: 3.722994883855184 - Train Bleu: 0.1440980457386363 - Valid Bleu: 0.13850288960178694
Epoch 8/50 - Loss: 3.5378352801005044 - Train Bleu: 0.142753354451392 - Valid Bleu: 0.13333011541535147
Epoch 9/50 - Loss: 3.490084727605184 - Train Bleu: 0.15401875324856093 - Valid Bleu: 0.1547107015402966
Epoch 10/50 - Loss: 3.429983139038086 - Train Bleu: 0.15825839384455223 - Valid Bleu: 0.17893309223530265
Epoch 11/50 - Loss: 3.315243403116862 - Train Bleu: 0.15884878466756258 - Valid Bleu: 0.17663644669500383
Epoch 12/50 - Loss: 3.2774600187937417 - Train Bleu: 0.16340826310991935 - Valid Bleu: 0.18621865951513666
Epoch 13/50 - Loss: 3.178585926691691 - Train Bleu: 0.1701378248250083 - Valid Bleu: 0.17676453222947405
Epoch 14/50 - Loss: 3.1456100145975747 - Train Bleu: 0.17129832247422921 - Valid Bleu: 0.1795874652842534
Epoch 15/50 - Loss: 3.0690738360087075 - Train Bleu: 0.17079925278367195 - Valid Bleu: 0.1900346454466967
Epoch 16/50 - Loss: 3.035691181818644 - Train Bleu: 0.1724214803323088 - Valid Bleu: 0.17969336951398623
Epoch 17/50 - Loss: 2.9812137285868325 - Train Bleu: 0.17695292958474843 - Valid Bleu: 0.18239902786313156
Epoch 18/50 - Loss: 2.9356561501820884 - Train Bleu: 0.18056820989090916 - Valid Bleu: 0.19017789510343625
Epoch 19/50 - Loss: 2.8590168158213296 - Train Bleu: 0.18690853487157566 - Valid Bleu: 0.19447821833058518
Epoch 20/50 - Loss: 2.837545871734619 - Train Bleu: 0.18906954093706763 - Valid Bleu: 0.1957500006722316
Epoch 21/50 - Loss: 2.780193646748861 - Train Bleu: 0.19650902970271156 - Valid Bleu: 0.2023447250697084
Epoch 22/50 - Loss: 2.685336192448934 - Train Bleu: 0.2069908427178476 - Valid Bleu: 0.2060849564374234
Epoch 23/50 - Loss: 2.688586155573527 - Train Bleu: 0.21587185315577342 - Valid Bleu: 0.22113314548942695
Epoch 24/50 - Loss: 2.634838819503784 - Train Bleu: 0.22409517861907824 - Valid Bleu: 0.23043606607198178
Epoch 25/50 - Loss: 2.589173952738444 - Train Bleu: 0.22924147806817474 - Valid Bleu: 0.22724529551219635
Epoch 26/50 - Loss: 2.5576323668162027 - Train Bleu: 0.23530089418869263 - Valid Bleu: 0.23197726520528475
Epoch 27/50 - Loss: 2.4715068340301514 - Train Bleu: 0.24638155910745277 - Valid Bleu: 0.2519681717401964
Epoch 28/50 - Loss: 2.413994789123535 - Train Bleu: 0.2557321643217043 - Valid Bleu: 0.2540551301257368
Epoch 29/50 - Loss: 2.387221654256185 - Train Bleu: 0.26658223118778934 - Valid Bleu: 0.2587837164240049
Epoch 30/50 - Loss: 2.3482062021891275 - Train Bleu: 0.275138173547821 - Valid Bleu: 0.2587125324521668
Epoch 31/50 - Loss: 2.2928949197133384 - Train Bleu: 0.2858664600407927 - Valid Bleu: 0.27044249163549233
Epoch 32/50 - Loss: 2.2164711157480874 - Train Bleu: 0.29930715809483976 - Valid Bleu: 0.2794641800824641
Epoch 33/50 - Loss: 2.1726065476735434 - Train Bleu: 0.3123439988591187 - Valid Bleu: 0.28290079014111114
Epoch 34/50 - Loss: 2.122920831044515 - Train Bleu: 0.32736347228086066 - Valid Bleu: 0.2887155304852844
Epoch 35/50 - Loss: 2.0636020501454673 - Train Bleu: 0.3390295587380819 - Valid Bleu: 0.3032577885941727
Epoch 36/50 - Loss: 1.995132287343343 - Train Bleu: 0.3624479012665716 - Valid Bleu: 0.3107978634085114
Epoch 37/50 - Loss: 1.9365613063176472 - Train Bleu: 0.3850295693626671 - Valid Bleu: 0.31060032614887245
Epoch 38/50 - Loss: 1.9295032819112141 - Train Bleu: 0.4053646422623167 - Valid Bleu: 0.31331619888815576
Epoch 39/50 - Loss: 1.8664948145548503 - Train Bleu: 0.4351268525465241 - Valid Bleu: 0.3261936193146199
Epoch 40/50 - Loss: 1.8112891515096028 - Train Bleu: 0.4604079882351191 - Valid Bleu: 0.3332334577199612
Epoch 41/50 - Loss: 1.761149287223816 - Train Bleu: 0.48209567680892096 - Valid Bleu: 0.3367688613568326
Epoch 42/50 - Loss: 1.7068832317988079 - Train Bleu: 0.49371772147269155 - Valid Bleu: 0.3414854952241741
Epoch 43/50 - Loss: 1.647416313489278 - Train Bleu: 0.5075018199510736 - Valid Bleu: 0.34973534565363157
Epoch 44/50 - Loss: 1.661431113878886 - Train Bleu: 0.5233347428747755 - Valid Bleu: 0.3493406632366331
Epoch 45/50 - Loss: 1.5958428382873535 - Train Bleu: 0.5318183734965426 - Valid Bleu: 0.357046153984349
Epoch 46/50 - Loss: 1.5041192372639973 - Train Bleu: 0.5370584973802871 - Valid Bleu: 0.3642419043387788
Epoch 47/50 - Loss: 1.487096627553304 - Train Bleu: 0.5493238473531351 - Valid Bleu: 0.3583331091490889
Epoch 48/50 - Loss: 1.4775272210439045 - Train Bleu: 0.5594084559729191 - Valid Bleu: 0.3586776351521222
Epoch 49/50 - Loss: 1.439125657081604 - Train Bleu: 0.5649402035060306 - Valid Bleu: 0.3604591564329621
Epoch 50/50 - Loss: 1.3888862133026123 - Train Bleu: 0.5729100702483385 - Valid Bleu: 0.3612841243512638
```



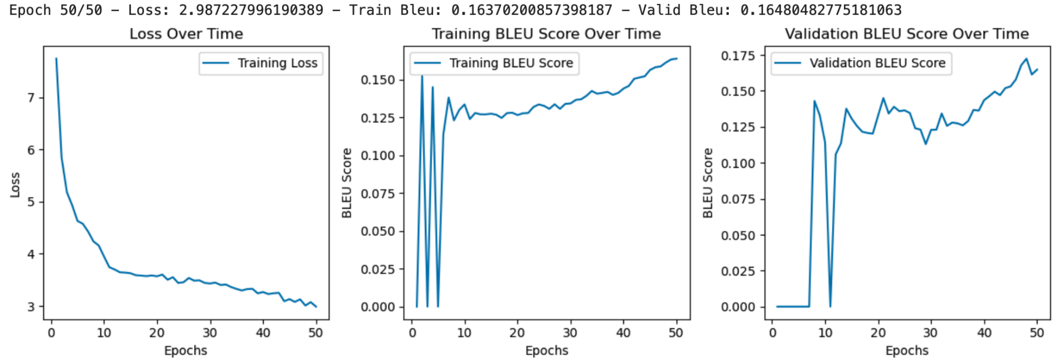

```
Training complete.
```

```
gru_rnn_model.eval()
prediction_with_translation_scores(gru_rnn_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
model_similarity_score(gru_rnn_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
```

```
Average BLEU Score: 0.3519
Average METEOR Score: 0.2994
Average Similarity Score 0.4683838710
```

Figure 13: GRU batch size = 1024 and epochs = 50

```
Epoch 1/50 - Loss: 7.740167140960693 - Train Bleu: 0 - Valid Bleu: 0
Epoch 2/50 - Loss: 5.841937065124512 - Train Bleu: 0.15234766028705451 - Valid Bleu: 0
Epoch 3/50 - Loss: 5.1786831219991045 - Train Bleu: 0 - Valid Bleu: 0
Epoch 4/50 - Loss: 4.927914301554362 - Train Bleu: 0.14490004674324472 - Valid Bleu: 0
Epoch 5/50 - Loss: 4.629237333933513 - Train Bleu: 0 - Valid Bleu: 0
Epoch 6/50 - Loss: 4.57489029566447 - Train Bleu: 0.113701814716755 - Valid Bleu: 0
Epoch 7/50 - Loss: 4.425365924835205 - Train Bleu: 0.1380464529464578 - Valid Bleu: 0
Epoch 8/50 - Loss: 4.238068580627441 - Train Bleu: 0.12293653479228328 - Valid Bleu: 0.14300485914708025
Epoch 9/50 - Loss: 4.159181118011475 - Train Bleu: 0.12982309900066327 - Valid Bleu: 0.1330111580025172
Epoch 10/50 - Loss: 3.9481235345204673 - Train Bleu: 0.13350251738697422 - Valid Bleu: 0.11419461902870459
Epoch 11/50 - Loss: 3.7434867223103843 - Train Bleu: 0.12383617204272256 - Valid Bleu: 0
Epoch 12/50 - Loss: 3.6989821592966714 - Train Bleu: 0.12781486420992494 - Valid Bleu: 0.10582550911038122
Epoch 13/50 - Loss: 3.6452163060506186 - Train Bleu: 0.12693651263203568 - Valid Bleu: 0.11356723976658731
Epoch 14/50 - Loss: 3.63926895459493 - Train Bleu: 0.12691863280060722 - Valid Bleu: 0.137647926014517
Epoch 15/50 - Loss: 3.6267836888631186 - Train Bleu: 0.1273552286742758 - Valid Bleu: 0.1308123600797337
Epoch 16/50 - Loss: 3.5890448888142905 - Train Bleu: 0.1266771715083212 - Valid Bleu: 0.12568930325479233
Epoch 17/50 - Loss: 3.5818671385447183 - Train Bleu: 0.12456908809244839 - Valid Bleu: 0.12162577293966437
Epoch 18/50 - Loss: 3.572046677271525 - Train Bleu: 0.1277567751474002 - Valid Bleu: 0.12074664845256405
Epoch 19/50 - Loss: 3.5844043095906577 - Train Bleu: 0.12801691886702776 - Valid Bleu: 0.12027272624298986
Epoch 20/50 - Loss: 3.568852504094442 - Train Bleu: 0.12645853104213023 - Valid Bleu: 0.13277776599641802
Epoch 21/50 - Loss: 3.601226329803467 - Train Bleu: 0.1276240758532193 - Valid Bleu: 0.14502153911698853
Epoch 22/50 - Loss: 3.5033982594807944 - Train Bleu: 0.12783455394308793 - Valid Bleu: 0.13426210762375612
Epoch 23/50 - Loss: 3.5506725311279297 - Train Bleu: 0.13176769219803014 - Valid Bleu: 0.1389831165887496
Epoch 24/50 - Loss: 3.4408464431762695 - Train Bleu: 0.13347316943704413 - Valid Bleu: 0.13591352472606594
Epoch 25/50 - Loss: 3.4549601078033447 - Train Bleu: 0.13254405669652977 - Valid Bleu: 0.13649653492784417
Epoch 26/50 - Loss: 3.5366575717926025 - Train Bleu: 0.1305514494964497 - Valid Bleu: 0.13461039541142591
Epoch 27/50 - Loss: 3.484767436981201 - Train Bleu: 0.133594152550766 - Valid Bleu: 0.12406313242601355
Epoch 28/50 - Loss: 3.4909427165985107 - Train Bleu: 0.1305746036715261 - Valid Bleu: 0.12309037794599072
Epoch 29/50 - Loss: 3.44195818901062 - Train Bleu: 0.13380945825870033 - Valid Bleu: 0.11296987191453345
Epoch 30/50 - Loss: 3.43056853612264 - Train Bleu: 0.13413427590607596 - Valid Bleu: 0.12292281116382663
Epoch 31/50 - Loss: 3.4472935994466147 - Train Bleu: 0.13648134255540953 - Valid Bleu: 0.12302774454543725
Epoch 32/50 - Loss: 3.402024825414022 - Train Bleu: 0.13689045353109566 - Valid Bleu: 0.1342714597065156
Epoch 33/50 - Loss: 3.410606543223063 - Train Bleu: 0.1391723598945073 - Valid Bleu: 0.12569952259802464
Epoch 34/50 - Loss: 3.3617519537607827 - Train Bleu: 0.14236094837526334 - Valid Bleu: 0.12800963072096236
Epoch 35/50 - Loss: 3.326902707417806 - Train Bleu: 0.14058727750835004 - Valid Bleu: 0.12741398143871724
Epoch 36/50 - Loss: 3.296213706334432 - Train Bleu: 0.14110767905542226 - Valid Bleu: 0.12603738362142067
Epoch 37/50 - Loss: 3.3235374291737876 - Train Bleu: 0.14173444728092924 - Valid Bleu: 0.1291045614579128
Epoch 38/50 - Loss: 3.328990936279297 - Train Bleu: 0.13978708836319934 - Valid Bleu: 0.1367800886037089
Epoch 39/50 - Loss: 3.241856892903646 - Train Bleu: 0.1410934479927366 - Valid Bleu: 0.13632256750861102
Epoch 40/50 - Loss: 3.2666401863098145 - Train Bleu: 0.14396218045075163 - Valid Bleu: 0.1435441801840794
Epoch 41/50 - Loss: 3.2288320859273276 - Train Bleu: 0.1457171185634293 - Valid Bleu: 0.1463852892127024
Epoch 42/50 - Loss: 3.2439067363739014 - Train Bleu: 0.15041651043757892 - Valid Bleu: 0.14949713032118864
Epoch 43/50 - Loss: 3.2507809003194175 - Train Bleu: 0.15126460615651513 - Valid Bleu: 0.14706059912079547
Epoch 44/50 - Loss: 3.0906397501627603 - Train Bleu: 0.15206507578913567 - Valid Bleu: 0.1519417565791037
Epoch 45/50 - Loss: 3.131195624669393 - Train Bleu: 0.15624025377623668 - Valid Bleu: 0.15316924513323138
Epoch 46/50 - Loss: 3.078008015950521 - Train Bleu: 0.15800965863660713 - Valid Bleu: 0.15800153300198844
Epoch 47/50 - Loss: 3.1265732447306314 - Train Bleu: 0.15859795795116446 - Valid Bleu: 0.1678217763356289
Epoch 48/50 - Loss: 3.0081962744394937 - Train Bleu: 0.16117166742037428 - Valid Bleu: 0.17243187459065829
Epoch 49/50 - Loss: 3.073342482248942 - Train Bleu: 0.16321381466133547 - Valid Bleu: 0.16139624417506185
Epoch 50/50 - Loss: 2.987227996190389 - Train Bleu: 0.16370200857398187 - Valid Bleu: 0.16480482775181063
```



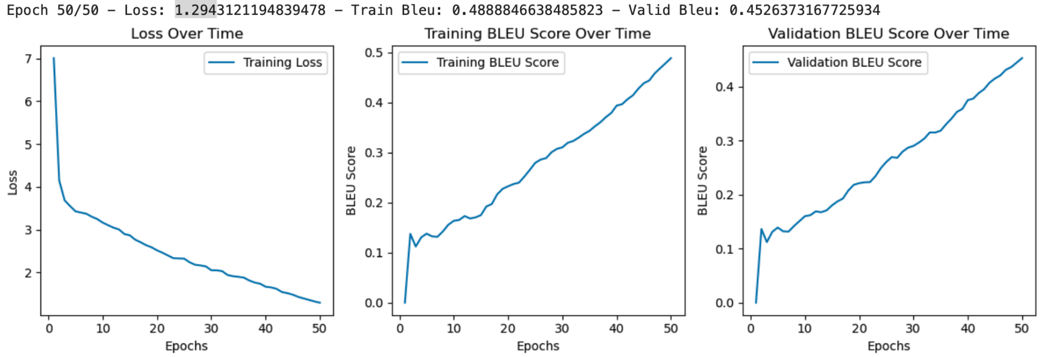

```
Training complete.
```

```
lstm_rnn_model.eval()
prediction_with_translation_scores(lstm_rnn_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
model_similarity_score(lstm_rnn_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
```

```
Average BLEU Score: 0.1681
Average METEOR Score: 0.1588
Average Similarity Score 0.3005668795
```

Figure 14: LSTM batch size = 1024 and epochs = 50

```
Epoch 1/50 - Loss: 7.008315086364746 - Train Bleu: 0 - Valid Bleu: 0
Epoch 2/50 - Loss: 4.149343490600586 - Train Bleu: 0.13758035913331867 - Valid Bleu: 0.13640560252892492
Epoch 3/50 - Loss: 3.6844462553660073 - Train Bleu: 0.1122218929656873 - Valid Bleu: 0.11215323251767581
Epoch 4/50 - Loss: 3.5515368779500327 - Train Bleu: 0.13052360345666858 - Valid Bleu: 0.13111486804812653
Epoch 5/50 - Loss: 3.4283974170684814 - Train Bleu: 0.13798757789352592 - Valid Bleu: 0.13906228240593296
Epoch 6/50 - Loss: 3.3997510274251304 - Train Bleu: 0.1325585764815879 - Valid Bleu: 0.13208366037729963
Epoch 7/50 - Loss: 3.3718620936075845 - Train Bleu: 0.13171892531581655 - Valid Bleu: 0.13157977524236966
Epoch 8/50 - Loss: 3.3025382359822593 - Train Bleu: 0.14251220549590402 - Valid Bleu: 0.14186859833153034
Epoch 9/50 - Loss: 3.2485945224761963 - Train Bleu: 0.15583240682857694 - Valid Bleu: 0.1509494461562389
Epoch 10/50 - Loss: 3.1658029556274414 - Train Bleu: 0.16365435353643967 - Valid Bleu: 0.16004721392599794
Epoch 11/50 - Loss: 3.1044824918111167 - Train Bleu: 0.16538418727177726 - Valid Bleu: 0.16189316091136388
Epoch 12/50 - Loss: 3.046987851460775 - Train Bleu: 0.17313890150674768 - Valid Bleu: 0.16897405448338912
Epoch 13/50 - Loss: 3.006123145421346 - Train Bleu: 0.1683574518712741 - Valid Bleu: 0.167415373234693
Epoch 14/50 - Loss: 2.898799498875936 - Train Bleu: 0.17045822125158652 - Valid Bleu: 0.17086988038703704
Epoch 15/50 - Loss: 2.867554187774658 - Train Bleu: 0.17501127565117502 - Valid Bleu: 0.18011463412119097
Epoch 16/50 - Loss: 2.7665205001831055 - Train Bleu: 0.19232481384289507 - Valid Bleu: 0.18745189611915164
Epoch 17/50 - Loss: 2.705775419871012 - Train Bleu: 0.1971415107148571 - Valid Bleu: 0.192688478563238
Epoch 18/50 - Loss: 2.637535889943441 - Train Bleu: 0.2166625034643793 - Valid Bleu: 0.20759155051285785
Epoch 19/50 - Loss: 2.588974634806315 - Train Bleu: 0.2276032284113171 - Valid Bleu: 0.21810309136251763
Epoch 20/50 - Loss: 2.518366018931071 - Train Bleu: 0.23268720020143355 - Valid Bleu: 0.22120459874203488
Epoch 21/50 - Loss: 2.4621567726135254 - Train Bleu: 0.23714934463266382 - Valid Bleu: 0.22274591436279334
Epoch 22/50 - Loss: 2.398523489634196 - Train Bleu: 0.23986417455167947 - Valid Bleu: 0.22319326263994219
Epoch 23/50 - Loss: 2.3346774578094482 - Train Bleu: 0.25202120252034077 - Valid Bleu: 0.23422080771464826
Epoch 24/50 - Loss: 2.329048236211141 - Train Bleu: 0.26548601320104215 - Valid Bleu: 0.24915206939594994
Epoch 25/50 - Loss: 2.3240249951680503 - Train Bleu: 0.27951610142227273 - Valid Bleu: 0.26051668456181476
Epoch 26/50 - Loss: 2.2387819290161133 - Train Bleu: 0.2860992895101952 - Valid Bleu: 0.26946409569571955
Epoch 27/50 - Loss: 2.181243975957235 - Train Bleu: 0.28889963841152705 - Valid Bleu: 0.2680358966908079
Epoch 28/50 - Loss: 2.1652355988820395 - Train Bleu: 0.3006926370269623 - Valid Bleu: 0.27962262641906
Epoch 29/50 - Loss: 2.142497936884562 - Train Bleu: 0.3075590080926086 - Valid Bleu: 0.28681160639438796
Epoch 30/50 - Loss: 2.0528053045272827 - Train Bleu: 0.310498427218099 - Valid Bleu: 0.29010648522073335
Epoch 31/50 - Loss: 2.0507869720458984 - Train Bleu: 0.3196299953900679 - Valid Bleu: 0.2963403566862461
Epoch 32/50 - Loss: 2.0317498445510864 - Train Bleu: 0.32323757568498007 - Valid Bleu: 0.3037479584864483
Epoch 33/50 - Loss: 1.9413164854049683 - Train Bleu: 0.329851808823339 - Valid Bleu: 0.31506167638831817
Epoch 34/50 - Loss: 1.9130910237630208 - Train Bleu: 0.3374940438817738 - Valid Bleu: 0.31491771182913597
Epoch 35/50 - Loss: 1.8984658320744832 - Train Bleu: 0.3434992404199521 - Valid Bleu: 0.3182276998257184
Epoch 36/50 - Loss: 1.8800524075826008 - Train Bleu: 0.35265006928716097 - Valid Bleu: 0.3301929308236764
Epoch 37/50 - Loss: 1.8129264911015828 - Train Bleu: 0.3606714914211601 - Valid Bleu: 0.34072388103252893
Epoch 38/50 - Loss: 1.7651193141937256 - Train Bleu: 0.3708528727929882 - Valid Bleu: 0.3530874322207837
Epoch 39/50 - Loss: 1.7374530633290608 - Train Bleu: 0.3791997526854384 - Valid Bleu: 0.35862904006732993
Epoch 40/50 - Loss: 1.667575478553772 - Train Bleu: 0.3939157794331564 - Valid Bleu: 0.3748325370092095
Epoch 41/50 - Loss: 1.6523573795954387 - Train Bleu: 0.39679664065793735 - Valid Bleu: 0.3776564685418906
Epoch 42/50 - Loss: 1.6190319061279297 - Train Bleu: 0.40676668808979227 - Valid Bleu: 0.3877747589177147
Epoch 43/50 - Loss: 1.5446185668309529 - Train Bleu: 0.4145978427192815 - Valid Bleu: 0.3950899979696524
Epoch 44/50 - Loss: 1.5174576044082642 - Train Bleu: 0.4279055509489762 - Valid Bleu: 0.407124121052915
Epoch 45/50 - Loss: 1.4822815259297688 - Train Bleu: 0.4387168967844698 - Valid Bleu: 0.41487129488347135
Epoch 46/50 - Loss: 1.430547873179118 - Train Bleu: 0.444135954637017 - Valid Bleu: 0.42062110452549584
Epoch 47/50 - Loss: 1.393683393796285 - Train Bleu: 0.458273152332482 - Valid Bleu: 0.4309679917497614
Epoch 48/50 - Loss: 1.3596947193145752 - Train Bleu: 0.4686380551970248 - Valid Bleu: 0.43603791986041657
Epoch 49/50 - Loss: 1.3239109913508098 - Train Bleu: 0.47858930972524616 - Valid Bleu: 0.4443210983129796
Epoch 50/50 - Loss: 1.2943121194839478 - Train Bleu: 0.4888846638485823 - Valid Bleu: 0.4526373167725934
```




```
Training complete.
```

```
gru_seq2seq_model.eval()
prediction_with_translation_scores_seq2(gru_seq2seq_model, test_sentences, test_fr_sentences, english_tokens, french_tokens
model_similarity_score_seq2(gru_seq2seq_model, test_sentences, test_fr_sentences, english_tokens, french_tokens)
```

```
Average BLEU Score: 0.4504
Average METEOR Score: 0.4799
Average Similarity Score 0.5787756658
```

Figure 15: GRU Seq2Seq batch size = 1024 and epochs = 50