Reinelle Jan Bugnot
G2304329L

I. Experiment Set-up

We begin with 14,574 English-to-Frensh translation data pairs from *tatoeba.org*. The original imported data contained 135,842 sentence pairs; however, in order to simply the training process for this assignment, the raw data was filtered to only include sentences with a maximum length of 15, and those beginning with a prefixes from a defined set (e.g. “He is...”, “I am...”). The main NLP task that this assignment will explore is machine translation, specifically, French to English, using 5 different encoder-decoder architectures, namely:

encoder-decoder

1. GRU-GRU (base model provided)
2. LSTM-LSTM
3. BiLSTM-GRU
4. GRU-GRU with Self-Attention
5. Transformer-GRU

All 5 models were trained for 10 epochs, and evaluated by measuring their performance using ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, metric. Rouge scores are typically used in evaluating the quality of machine-generated translations, with Rouge-1 measuring the overlap of unigrams (individual words) and Rouge-2 on bigrams (consecutive pairs of words). *Precision*, a key metric in this evaluation, quantifies the ratio of shared n-grams between the predicted translation and the ground truth text to the total number of n-grams in the translation. *Recall*, on the other hand, measures the ratio of shared n-grams to the total number of n-grams in the ground truth text. Lastly, *F-measure*, a composite metric combining precision and recall, produces comprehensive comparisons across experimental outcomes.

II. Discussion of Results

TABLE I. Rouge Scores of all 5 models on the TRAINING SET

	Rouge 1			***Rouge 2***		
	F-measure	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**
GRU-GRU	**0.8819725**	**0.812354**	**0.9675259**	**0.8312137**	**0.7519789**	**0.934447**
LSTM-LSTM	0.8742195	0.805055	0.9591293	0.8186284	0.7405914	0.920091
BiLSTM-GRU	0.8703530	0.800595	0.9565667	0.8076291	0.7294545	0.909982
GRU-GRU+SA	0.8429847	0.776945	0.9255562	0.7581787	0.6856827	0.854412
Transformer-GRU	0.1816590	0.214643	0.1665801	0.0948355	0.1160059	0.087648

TABLE II. Rouge Scores of all 5 models on the TEST SET

	Rouge 1			***Rouge 2***		
	F-measure	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**
GRU-GRU	**0.6795638**	**0.6317817**	**0.7431425**	**0.5086816**	**0.4638413**	**0.5712976**
LSTM-LSTM	0.6629833	0.6157039	0.7250851	0.4957904	0.45141014	0.5566465
BiLSTM-GRU	0.668898	0.6185983	0.7360021	0.4941841	0.44795063	0.5588633
GRU-GRU+SA	0.645760	0.5997530	0.7076982	0.4676110	0.42554376	0.5268964
Transformer-GRU	0.175393	0.2068587	0.1609372	0.0898926	0.10888203	0.0834802

* *SA: Self-Attention*

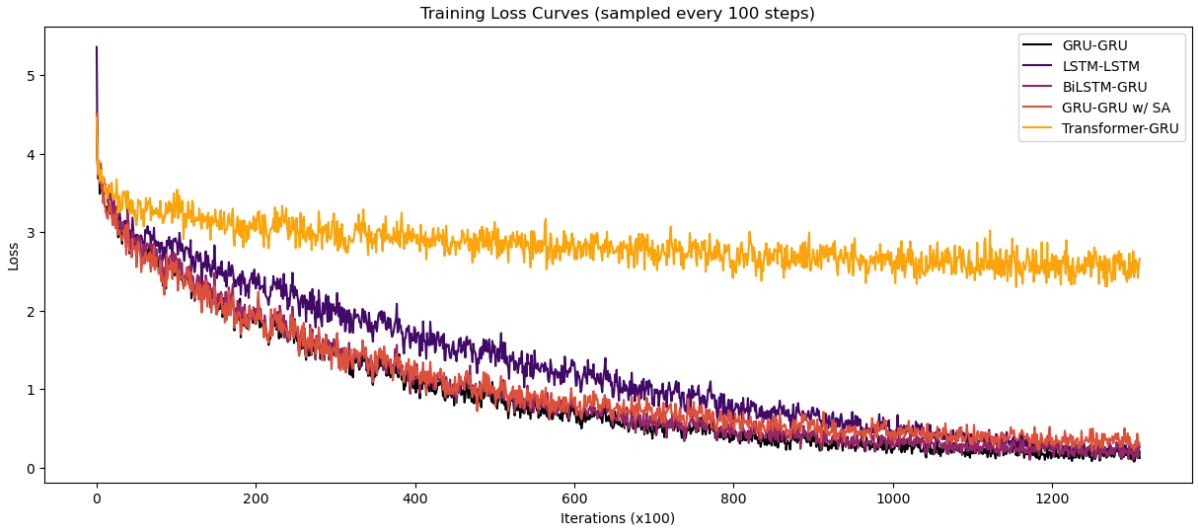

Fig. 1. Training loss curves for all 5 models across 10 epochs

GRU-GRU outperforms all other configurations

Table I and Table II presents the summary results of the Rouge 1 and 2 scores, while Fig. 1 shows the training loss curves generated across 10 epochs for all 5 models. Here, we can see that the GRU-GRU (base) model outperformed all other configurations in all evaluation criteria. This is likely due to its simplicity and efficiency.

We can also observe that the LSTM-LSTM and BiLSTM-GRU configurations produced results that compete very closely with the GRU-GRU. It is possible that by tuning the hyperparameters of the LSTM / BiLSTM models better instead of using the default values, we can potentially get results that outperform GRU-GRU. In general, GRUs (or Gated Recurrent Units) are known for their similar performance as LSTMs (or Long Short-Term Memory) networks, but having the advantage of being less complex. On smaller datasets, GRUs sometimes offer better results (Chung et al., 2014). Given that the dataset used for this assignment is relatively small (around 14.5k samples), it is therefore reasonable to observe GRU-GRU outperforming more complex configurations.

Self-Attention's slight underperformance

Another notable observation in the experiments is that the addition of self-attention to the GRU-GRU base model produced worse performance, which can be surprising considering self-attention was one of the bigger breakthroughs in the field of NLP. This could possibly be attributed to the small dataset size. Attention mechanisms often require larger datasets to generalize effectively (Vaswani et al., 2017). The performance of self-attention mechanisms can also be sensitive to hyperparameter settings. Factors such as the number of attention heads, the dimensionality of the model, and the learning rate can significantly impact the model's performance. If these hyperparameters are not optimally tuned, the model might underperform. In the case of this assignment, I reused the same hyperparameters (hidden layer size of 512) for all 5 models, in an attempt to standardize the results. However, since different models process data in completely different ways, then each model likely require their own hyperparameter tuning to achieve proper (if not the maximum) convergence.

Transformer's extreme underperformance

Transformers are known to be data-hungry models. They were designed to handle large-scale tasks and can perform poorly when trained on smaller datasets (Vaswani et al., 2017). In this case, the dataset size of 14.5k pairs might not be sufficient for a Transformer model to learn effectively. This results in poor Rouge scores as seen in Table I and II, as well as poor convergence as shown in Fig. 1.

However, one other possibly notable observation is that the training and testing results of the Transformer model are relatively closer to each other (e.g., a training and testing rouge 1 scores for f-measure of 0.1816590 and 0.175393, respectively) compared to other models, which could indicate that the Transformer model is generalizing on unseen data much better than other models. Overfitting occurs when a model learns the training

data too well, to the point where it performs poorly on unseen data. The closer performance on both training and testing sets could suggest that the Transformer model is not overfitting, despite its overall lower performance.

Another potential reason for the underperformance of this configuration is that we used a GRU decoder on a Transformer encoder. Transformer encoders are powerful and can create rich context-aware representations. However, the GRU decoder might not have the capacity to fully utilize these representations. Transformers, with their self-attention mechanism, can capture complex patterns and long-range dependencies in the data. On the other hand, GRUs, while effective, might not be as capable in handling such complex representations, leading to a potential mismatch. The absence of an attention mechanism in the decoding phase could also be a big factor. In a full Transformer model, the decoder also has self-attention layers which help it focus on different parts of the input sequence while generating the output. This can be especially useful in tasks like machine translation where alignment between parts of the input and output sequences is important. By using a GRU decoder, we might be missing out on these benefits.

Training Times

TABLE III. Training times for all 5 models on 10 epochs

	GRU-GRU	**LSTM-LSTM**	**BiLSTM-GRU**	**GRU-GRU w/ SA**	**Transformer-GRU**
Training Time	29m 7s	31m 28s	48m 58s	35m 51s	50m 19s

Table III presents the amount of time it took to train each of the models in this assignment. Here, we can see that the Transformer-GRU model took the longest time to train. This could be due to the high computational complexity of Transformers, especially when combined with RNNs like GRU. Transformers involve matrix multiplications which are computationally expensive operations. Moreover, the self-attention mechanism in Transformers involves calculating attention scores for every pair of words in the input, which can be time-consuming.

Meanwhile, the GRU-GRU model had the shortest training time. GRUs are simpler than LSTMs and Transformers, and thus, they are computationally less expensive. They have fewer parameters and simpler gating mechanisms, which leads to faster training times.

References

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv preprint arXiv:1412.3555.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).