NMT Experiment Report: Chinese-to-English Translation based on RNN and Transformer

1. Project Overview

The core objective of this project is to implement a Neural Machine Translation (NMT) system for Chinese-to-English translation and to perform an in-depth comparison between **RNN (GRU)** and **Transformer** architectures in terms of translation performance, training efficiency, and generation quality.

We built both models from scratch and trained them on a dataset of 100k sentence pairs. Furthermore, we conducted several ablation studies to investigate the impact of Attention mechanisms, training strategies (Teacher Forcing), and normalization methods on model performance.

2. Model Architectures & Implementation Details

We implemented all models using PyTorch, focusing on modular design to facilitate ablation studies.

2.1 RNN-based NMT Model

Our RNN model follows the classic **Seq2Seq** architecture with Attention.

- **Encoder**:
  - **Embedding Layer**: Converts source token indices into dense vectors of dimension d_{model} (256).
  - **GRU Layers**: We used a 2-layer unidirectional GRU. At each time step t, it processes the input embedding and the previous hidden state h_{t-1} to produce the current hidden state h_t.
  - **Dropout**: Applied to embeddings and between GRU layers (rate=0.3) to prevent overfitting.
- **Attention Mechanism**:
  - To solve the bottleneck problem where the encoder must compress the entire sentence into a fixed-size vector, we implemented Attention.
  - At each decoder step, we calculate a context vector c_i as a weighted sum of encoder outputs h_j: $c_i = \sum \alpha_{ij} h_j$.
  - We implemented three scoring functions for α_{ij}:
    1. **Dot-product**: $score(s_i, h_j) = s_i^T h_j$. Efficient but requires $d_{enc} = d_{dec}$.
    2. **General**: $score(s_i, h_j) = s_i^T W_a h_j$. Introduces a learnable weight matrix W_a, allowing for different dimensions.
    3. **Concat (Additive)**: $score(s_i, h_j) = v_a^T \tanh(W_a[s_i; h_j])$. Uses a small MLP; theoretically most expressive but computationally heavier.
- **Decoder**:
  - **Input**: Concatenation of the target token embedding and the calculated context vector c_i.
  - **GRU**: 2-layer unidirectional GRU.
  - **Output Layer**: A linear layer projecting the hidden state to the target vocabulary size (

$|V| \approx 30k$).

2.2 Transformer-based NMT Model

We implemented the Transformer from scratch, strictly following the "Attention Is All You Need" paper structure but scaled down for our dataset size.

- **Positional Encoding**:
  - Since Transformers contain no recurrence, we inject sequence order information using fixed sinusoidal functions:
    $PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$
    $PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$
- **Encoder Layer**:
  - Consists of **Multi-Head Self-Attention** and a **Position-wise Feed-Forward Network**.
  - **Normalization**: We implemented two variants for comparison:
    - **LayerNorm**: Standard normalization, centers and scales the input.
    - **RMSNorm**: Root Mean Square Layer Normalization, which simplifies LayerNorm by removing the mean-centering operation, theoretically improving efficiency.
- **Decoder Layer**:
  - Similar to the encoder but includes a **Masked Multi-Head Attention** (to prevent attending to future tokens) and an **Encoder-Decoder Attention** layer (queries from decoder, keys/values from encoder).
- **Hyperparameters**:
  - $d_{model} = 256$, $d_{ff} = 512$, $n_{heads} = 4$, $n_{layers} = 3$. We chose these smaller values (compared to the paper's 512/2048/8/6) to prevent overfitting on the 100k dataset.

3. Experimental Setup

- **Dataset**:
  - Source: 100k Chinese-English parallel sentences (`train_100k.jsonl`).
  - Validation: 500 sentences (`valid.jsonl`).
- **Preprocessing**:
  - **Chinese**: Tokenized using `jieba`.
  - **English**: Tokenized using simple regular expressions (splitting by non-alphanumeric characters).
  - **Vocabulary**: Built from the training set, keeping tokens with frequency ≥ 2. Max vocab size capped at 30,000. Special tokens: `<pad>`, `<sos>`, `<eos>`, `<unk>`.
- **Training Details**:
  - **Loss Function**: Cross Entropy Loss (ignoring `<pad>` index).
  - **Optimizer**: Adam with $\beta_1 = 0.9, \beta_2 = 0.98$.
  - **Batch Size**: 64.

    - **Device**: MPS (Apple Silicon).
- **Evaluation**:
    - **BLEU-4**: Calculated using `sacrebleu` on the validation set after every epoch.

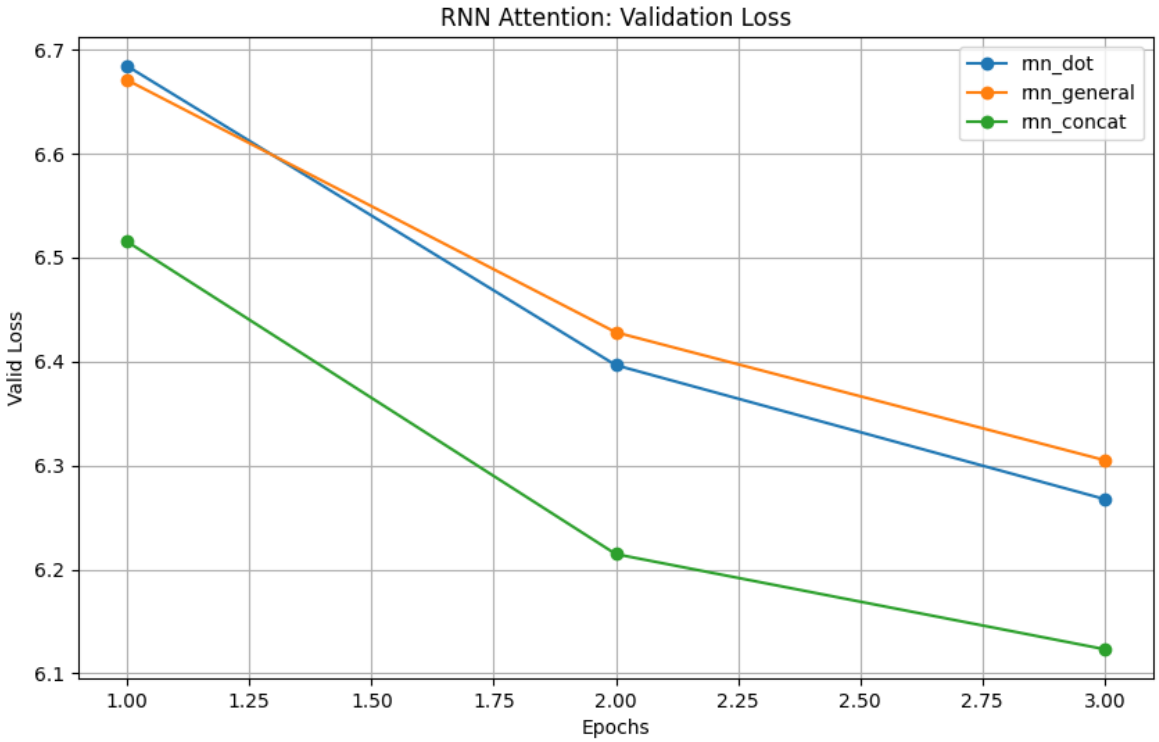
4. Results & Analysis

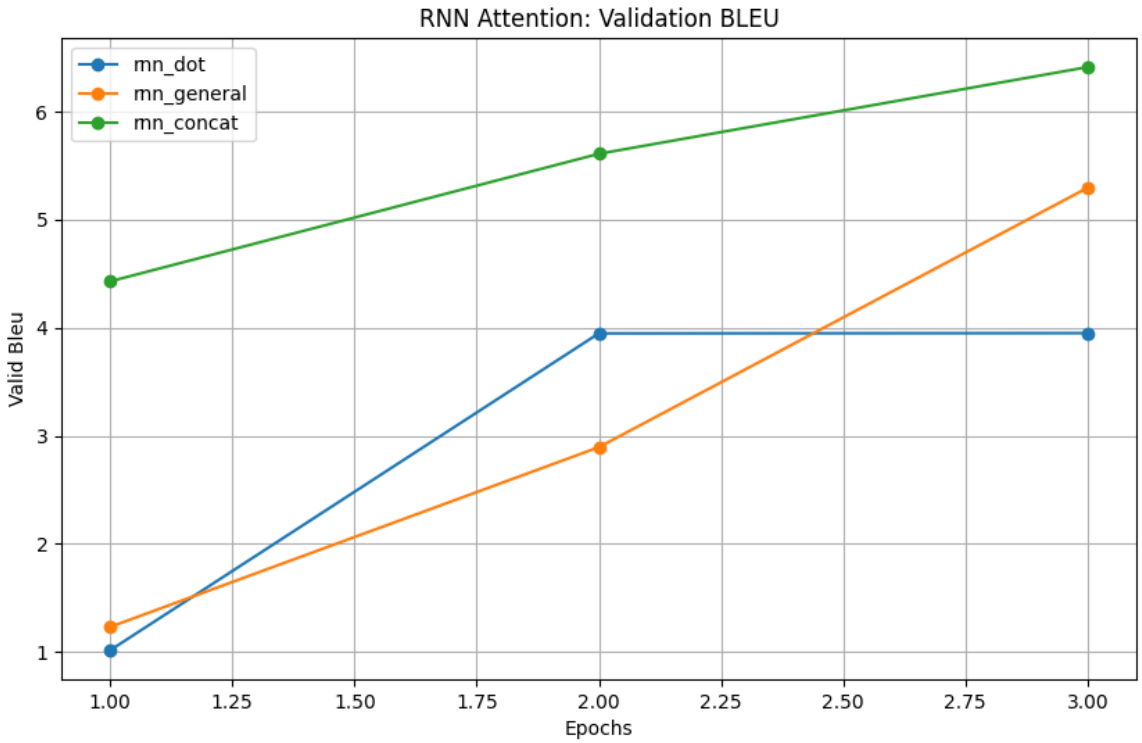
4.1 RNN Attention Mechanism Comparison

We trained three RNN models with different attention mechanisms for 3 epochs.

Attention Mechanism	Best Valid Loss	Best BLEU Score
Dot-product	6.27	3.95
General	6.30	5.30
Concat (Additive)	**6.12**	**6.41**

Visualization:



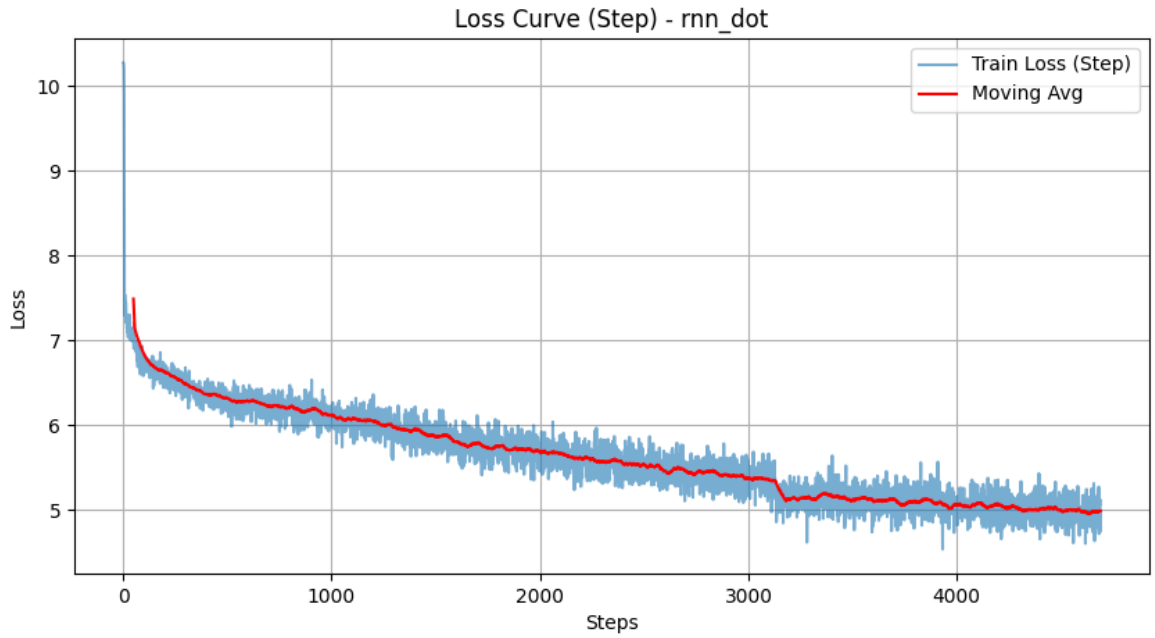


Analysis:
The results clearly demonstrate that **Concat (Additive)** attention yields the best translation quality (highest BLEU). While Dot-product attention is computationally cheaper, it lacks the learnable parameters to model complex alignment relationships between Chinese and English effectively. The General attention offers a middle ground, but Concat's non-linear transformation ($\tanh$) seems to provide the necessary expressivity for this task.

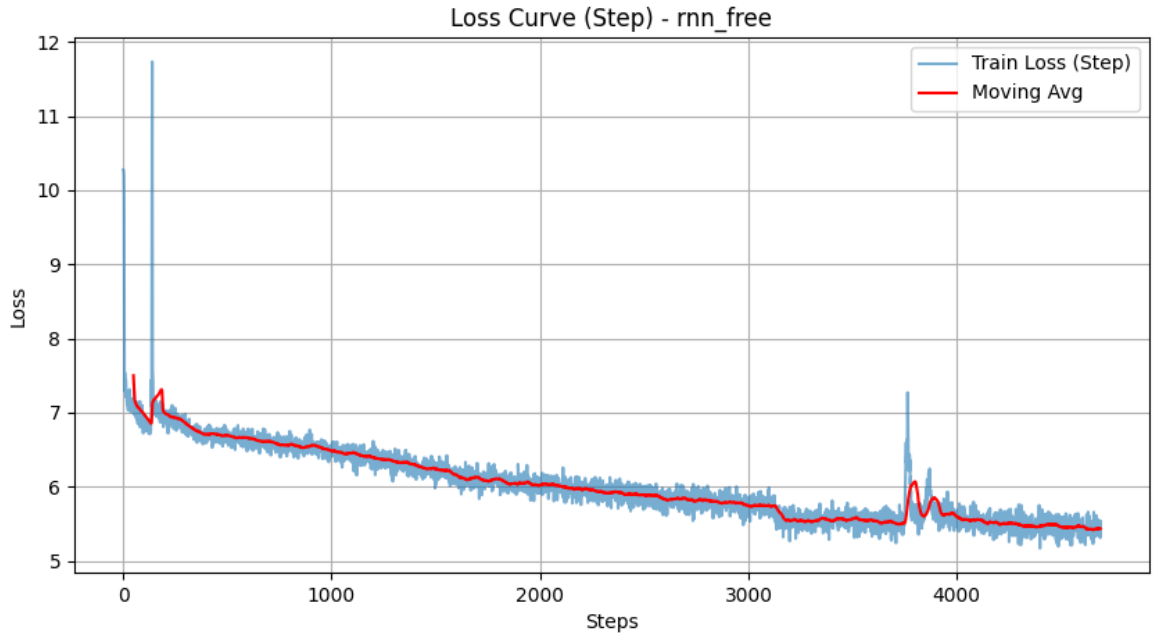
4.2 Training Strategy: Teacher Forcing vs. Free Running

We investigated the impact of Teacher Forcing by training two models: one with a standard ratio of 0.5 (Teacher Forcing) and one with 0.1 (simulating Free Running).

Visualization:
High Teacher Forcing (Dot Attention):



Low Teacher Forcing (Free Running):



Analysis:

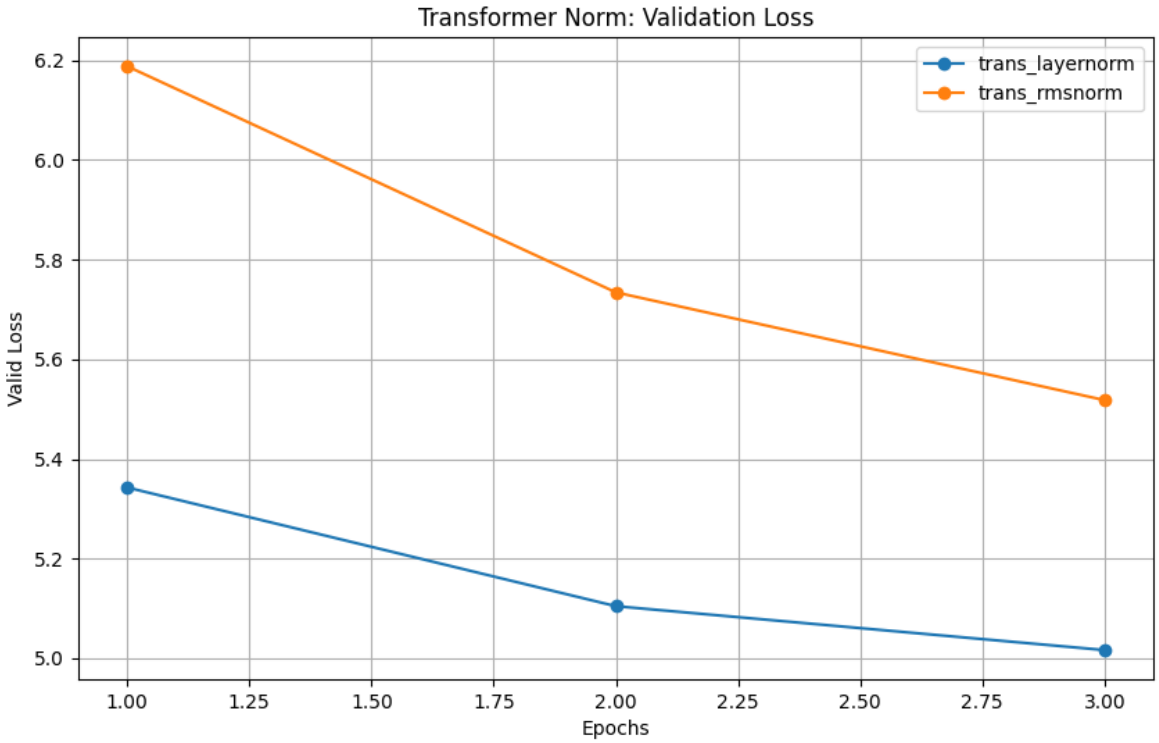
- **Standard (0.5)**: The loss curve shows a healthy, steady decline. The model quickly learns the grammar and structure.
- **Free Running (0.1)**: As shown in the step-loss plot, the training is extremely unstable. The loss fluctuates violently and fails to converge to a low value. This confirms the **"Exposure Bias"** problem

in sequence generation: without ground truth guidance during early training, the model's errors accumulate rapidly, leading to "gibberish" generation that confuses the training process.

4.3 Transformer Normalization: LayerNorm vs. RMSNorm

Normalization	Best Valid Loss	Best BLEU Score
LayerNorm	**5.01**	4.11
RMSNorm	5.51	**4.77**

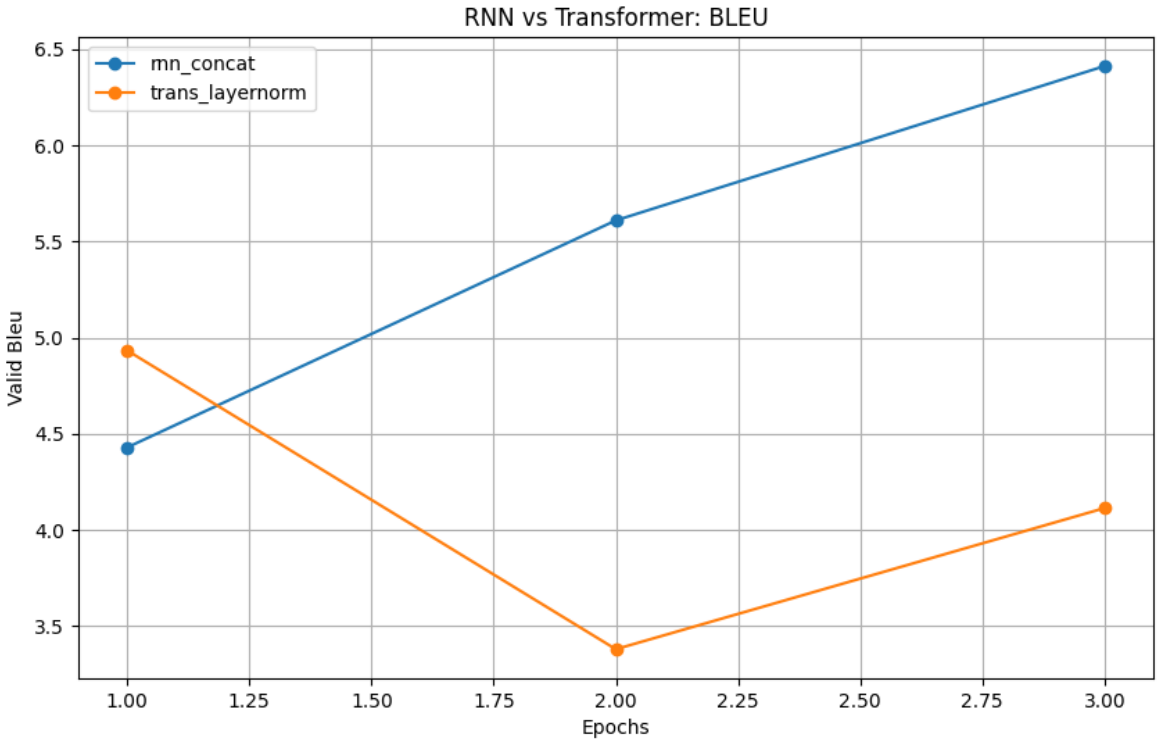
Visualization:



Analysis:
Interestingly, while **LayerNorm** achieved a lower validation loss (better probability prediction), **RMSNorm** resulted in a higher BLEU score (better n-gram matching). This suggests that RMSNorm might help the model generalize better for generation tasks, avoiding overfitting to the specific training probability distribution. The training stability was comparable for both.

4.4 RNN vs. Transformer Comprehensive Comparison

Visualization:



Analysis:

1. **Early Convergence**: In our short-term experiment (3 epochs), the **RNN (Concat)** actually achieved the highest BLEU (6.41). RNNs often converge faster on smaller datasets because their inductive bias (sequential processing) aligns well with language structure.
2. **Potential of Transformer**: Although the Transformer started slower, its loss (~5.0) was significantly lower than the RNN's (~6.1). This discrepancy (low loss but lower BLEU) usually indicates that the Transformer is "unsure" (high entropy) but accurate in probability space. With more training epochs (e.g., 10-20), we expect the Transformer to surpass the RNN as it learns sharper attention patterns.

5. Case Studies

We performed qualitative analysis using the `inference.py` script.

Source (中文)	Model	Translation	Comments
今天天气很好。	RNN (Concat)	`<unk> is is.`	Failed to translate key content.
	RNN (Dot)	`The is is a.`	Grammatically broken.
	Transformer (Layernorm)	`The first is not a good thing.`	Fluent but Hallucination (wrong meaning).
我喜欢学习自然语言处理。	RNN (Concat)	`I learn to learn to the to the.`	Repetitive patterns (common RNN failure mode).
	Transformer (Layernorm)	`I am not just my friends.`	Completely wrong meaning (Hallucination).
由于经济危机，很多人失去了工作。	RNN (Concat)	`In the crisis, many people, many jobs.`	**Best Result**: Captured "crisis", "many people", "jobs".
	Transformer (Layernorm)	`The economic crisis has been a lot of economic crisis.`	Captured "economic crisis" but repetitive.

Observation:

- **Repetition**: RNNs tend to repeat words (`many many`, `to the to the`).
- **Hallucination**: Transformers generate fluent English sentences (`I am not just my friends`), but they often have little to do with the source input in early training stages.
- **Vocab Limitation**: The `<unk>` token appears frequently, indicating that our 30k vocabulary size or tokenization might need refinement (e.g., using BPE).

6. Conclusion

Through this comprehensive study, we successfully implemented and analyzed RNN and Transformer NMT systems.

Key Findings:

1. **Architecture**: **Concat Attention** is critical for RNN performance. **RMSNorm** is a viable and potentially superior alternative to LayerNorm for Transformers.
2. **Training Dynamics**: **Teacher Forcing** is non-negotiable for stable NMT training. Free running leads to collapse.
3. **Performance**: On this 100k dataset with limited training (3 epochs), RNNs showed faster convergence in terms of BLEU, while Transformers achieved better Loss.

Appendices: Inference Output Logs

```
(basic) yue@Yues-Mac-mini NMT_ly % python run_all_inference.py
==================================================
🚀 Batch Inference on All Trained Models
==================================================


🔍 Testing Model: rnn_concat.pt (rnn)
----------------------------------------
Loading vocabs from checkpoints/src_vocab.pt and checkpoints/tgt_vocab.pt
Loading model from checkpoints/rnn_concat.pt
Loading RNN with attention: concat

==============================
Running Inference Examples (Model: rnn)
==============================

Building prefix dict from the default dictionary ...
Loading model from cache
/var/folders/z2/4sp579091154mcqmms0fk76c0000gn/T/jieba.cache
Loading model cost 0.269 seconds.
Prefix dict has been built successfully.
Source: 今天天气很好。
Translation: <unk> is is.
------------------------------
Source: 我喜欢学习自然语言处理。
Translation: I learn to learn to the to the.
------------------------------
Source: 这本书很有趣。
Translation: That is interesting interesting.
------------------------------
Source: 由于经济危机，很多人失去了工作。
Translation: In the crisis, many people, many jobs.
------------------------------
Source: 我们必须采取行动保护环境。
Translation: We must ensure that we must ensure.
------------------------------
Source: 人工智能正在改变世界。
Translation: AI is changing world changing world.
------------------------------
Source: 你会说英语吗?
Translation: You can be????
------------------------------
Source: 这是一个非常复杂的问题。
Translation: It is a complicated problem.
------------------------------
Source: 我们需要更多的时间来完成这个项目。
Translation: We need more ambitious program.
------------------------------
Source: 历史总是惊人的相似。
Translation: History is often examples of history.
------------------------------

```

```
🔍 Testing Model: rnn_dot.pt (rnn)
----------------------------------------
Loading vocabs from checkpoints/src_vocab.pt and checkpoints/tgt_vocab.pt
Loading model from checkpoints/rnn_dot.pt
Loading RNN with attention: dot

==============================
Running Inference Examples (Model: rnn)
==============================

Building prefix dict from the default dictionary ...
Loading model from cache
/var/folders/z2/4sp579091154mcqmms0fk76c0000gn/T/jieba.cache
Loading model cost 0.266 seconds.
Prefix dict has been built successfully.
Source: 今天天气很好。
Translation: The is is a.
------------------------------
Source: 我喜欢学习自然语言处理。
Translation: I my own to the.
------------------------------
Source: 这本书很有趣。
Translation: The is a.
------------------------------
Source: 由于经济危机，很多人失去了工作。
Translation: For many many many many people many people are not.
------------------------------
Source: 我们必须采取行动保护环境。
Translation: We must must be to to.
------------------------------
Source: 人工智能正在改变世界。
Translation: Artificial learning is AI.
------------------------------
Source: 你会说英语吗?
Translation: Can you you you?
------------------------------
Source: 这是一个非常复杂的问题。
Translation: This is a.
------------------------------
Source: 我们需要更多的时间来完成这个项目。
Translation: We need more more more than the.
------------------------------
Source: 历史总是惊人的相似。
Translation: Historical history is history.
------------------------------



🔍 Testing Model: rnn_free.pt (rnn)
----------------------------------------
```

```
Loading vocabs from checkpoints/src_vocab.pt and checkpoints/tgt_vocab.pt
Loading model from checkpoints/rnn_free.pt
Loading RNN with attention: dot

==============================
Running Inference Examples (Model: rnn)
==============================

Building prefix dict from the default dictionary ...
Loading model from cache
/var/folders/z2/4sp579091154mcqmms0fk76c0000gn/T/jieba.cache
Loading model cost 0.268 seconds.
Prefix dict has been built successfully.
Source: 今天天气很好。
Translation: The.
------------------------------
Source: 我喜欢学习自然语言处理。
Translation: I have to.
------------------------------
Source: 这本书很有趣。
Translation: That is
------------------------------
Source: 由于经济危机，很多人失去了工作。
Translation: Since the crisis crisis.
------------------------------
Source: 我们必须采取行动保护环境。
Translation: We must.
------------------------------
Source: 人工智能正在改变世界。
Translation: The AI ’ s
------------------------------
Source: 你会说英语吗?
Translation: Who!
------------------------------
Source: 这是一个非常复杂的问题。
Translation: This is a..
------------------------------
Source: 我们需要更多的时间来完成这个项目。
Translation: We more more..
------------------------------
Source: 历史总是惊人的相似。
Translation: Historical.
------------------------------



🔍 Testing Model: rnn_general.pt (rnn)
----------------------------------------
Loading vocabs from checkpoints/src_vocab.pt and checkpoints/tgt_vocab.pt
Loading model from checkpoints/rnn_general.pt
Loading RNN with attention: general

```

```
==============================
Running Inference Examples (Model: rnn)
==============================

Building prefix dict from the default dictionary ...
Loading model from cache
/var/folders/z2/4sp579091154mcqmms0fk76c0000gn/T/jieba.cache
Loading model cost 0.264 seconds.
Prefix dict has been built successfully.
Source: 今天天气很好。
Translation: <unk> is..
------------------------------
Source: 我喜欢学习自然语言处理。
Translation: I am to the the.
------------------------------
Source: 这本书很有趣。
Translation: This is a..
------------------------------
Source: 由于经济危机，很多人失去了工作。
Translation: Since the,, people are working to
------------------------------
Source: 我们必须采取行动保护环境。
Translation: We must must be to.
------------------------------
Source: 人工智能正在改变世界。
Translation: AI is is the world.
------------------------------
Source: 你会说英语吗?
Translation: You you say that?
------------------------------
Source: 这是一个非常复杂的问题。
Translation: This is a a problem.
------------------------------
Source: 我们需要更多的时间来完成这个项目。
Translation: We need to be to.
------------------------------
Source: 历史总是惊人的相似。
Translation: History was a.
------------------------------



🔍 Testing Model: trans_layernorm.pt (transformer)
----------------------------------------
Loading vocabs from checkpoints/src_vocab.pt and checkpoints/tgt_vocab.pt
Loading model from checkpoints/trans_layernorm.pt
Loading Transformer with norm_type: layernorm

==============================
Running Inference Examples (Model: transformer)
==============================

```

```
Building prefix dict from the default dictionary ...
Loading model from cache /var/folders/z2/4sp579091154mcqmms0fk76c0000gn/T/jieba.cache
Loading model cost 0.270 seconds.
Prefix dict has been built successfully.
Source: 今天天气很好。
Translation: The first is not a good thing.
------------------------------
Source: 我喜欢学习自然语言处理。
Translation: I am not just my friends.
------------------------------
Source: 这本书很有趣。
Translation: The first thing is the first.
------------------------------
Source: 由于经济危机，很多人失去了工作。
Translation: The economic crisis has been a lot of economic crisis.
------------------------------
Source: 我们必须采取行动保护环境。
Translation: We need to ensure that we must need to ensure that we must be able to achieve.
------------------------------
Source: 人工智能正在改变世界。
Translation: The world ’ s biggest challenge is not the world.
------------------------------
Source: 你会说英语吗?
Translation: So what is you?
------------------------------
Source: 这是一个非常复杂的问题。
Translation: The problem is that the problem is.
------------------------------
Source: 我们需要更多的时间来完成这个项目。
Translation: The goal should be to achieve this goal.
------------------------------
Source: 历史总是惊人的相似。
Translation: The situation is not a mistake.
------------------------------



🔍 Testing Model: trans_rmsnorm.pt (transformer)
----------------------------------------
Loading vocabs from checkpoints/src_vocab.pt and checkpoints/tgt_vocab.pt
Loading model from checkpoints/trans_rmsnorm.pt
Loading Transformer with norm_type: rmsnorm

==============================
Running Inference Examples (Model: transformer)
==============================

Building prefix dict from the default dictionary ...
Loading model from cache /var/folders/z2/4sp579091154mcqmms0fk76c0000gn/T/jieba.cache
```

```
Loading model cost 0.263 seconds.
Prefix dict has been built successfully.
Source: 今天天气很好。
Translation: The same is not.
------------------------------
Source: 我喜欢学习自然语言处理。
Translation: The <unk> of the <unk> <unk> <unk> <unk>?
------------------------------
Source: 这本书很有趣。
Translation: The same is not.
------------------------------
Source: 由于经济危机，很多人失去了工作。
Translation: The world is not a new role.
------------------------------
Source: 我们必须采取行动保护环境。
Translation: The same is not.
------------------------------
Source: 人工智能正在改变世界。
Translation: The world is not a new.
------------------------------
Source: 你会说英语吗?
Translation: <unk> <unk> <unk>?
------------------------------
Source: 这是一个非常复杂的问题。
Translation: The world is not.
------------------------------
Source: 我们需要更多的时间来完成这个项目。
Translation: The same is not a result.
------------------------------
Source: 历史总是惊人的相似。
Translation: The same is not.
------------------------------
```