# The Fourth Report of Deep Learning for Natural Language Processing

Yue Zhao
zy2303607@buaa.edu.cn

## Abstract

This report delves into the Seq2Seq and Transformer models, prominent in the field of Deep Learning for Natural Language Processing (NLP). These models revolutionize tasks such as machine translation, text summarization, and others by utilizing unique neural network architectures. The Seq2Seq model employs an Encoder-Decoder structure with LSTM networks to effectively handle long sequence data, addressing challenges like vanishing gradients. However, it struggles with processing speed and complexity due to its sequential nature. Conversely, the Transformer model utilizes an attention mechanism to enhance parallel data processing, improving efficiency and scalability significantly. Despite its advantages, the Transformer requires substantial computational resources, particularly in terms of memory when trained on large datasets. This paper explores these architectures, their mechanisms, applications, comparative advantages, and limitations in handling sequence transformation tasks.

## Introduction

**Seq2Seq:** The term Seq2Seq, short for Sequence to Sequence, refers to a model structure composed of an Encoder-Decoder architecture that was introduced in 2014. This model is primarily used in applications such as machine translation, question-answering systems, audio transcription, and image description, representing one of the most powerful concepts in deep learning. In a Seq2Seq model, both the input and the output are sequences, designed to perform sequence transformation tasks. Specifically, in the Seq2Seq architecture, the encoder encodes all input sequences into a unified semantic vector, referred to as the Context vector. This is then decoded by the decoder. During the decoding process, the output from the previous moment is continuously used as the input for the next, in a cyclic decoding process, until a stop signal is produced, as shown in **Figure 1**. Notably, while the lengths of the input and output sequences can vary, the semantic vector produced in the intermediary process remains of a fixed length.

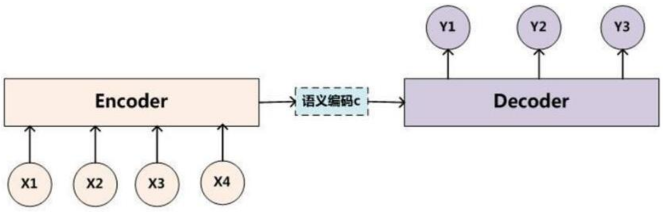

**Figure 1**: The structure of Seq2Seq model

The fundamental concept of the common encoder-decoder architecture involves utilizing two Recurrent Neural Networks (RNNs), one serving as the encoder and the other as the decoder. RNNs are a class of neural networks designed for processing sequential data. Due to their internal loops, they can maintain an internal state or memory, enabling them to handle temporal relationships in input data. However, RNNs face challenges when processing long sequence data, such as issues of vanishing gradients or exploding gradients, which make it difficult for the network to learn long-distance dependencies. To address these challenges, researchers have proposed several improved RNN structures, such as Long Short-Term Memory networks (LSTM) and Gated Recurrent Units (GRU). These architectures introduce gating mechanisms to regulate the flow of information, thereby effectively capturing long-term dependencies.

Among these, the Long Short-Term Memory (LSTM) network is a specialized type of RNN that is distinguished by its internal memory cells, which can store long-term information, as shown in **Figure 2**. This information is regulated through three types of gates: input gates, forget gates, and output gates. The gating mechanism enables the LSTM to retain significant information over long sequences while discarding irrelevant data, thus effectively managing dependencies in sequences of short to medium length. This makes LSTMs particularly suited for time series data and sequence prediction tasks.

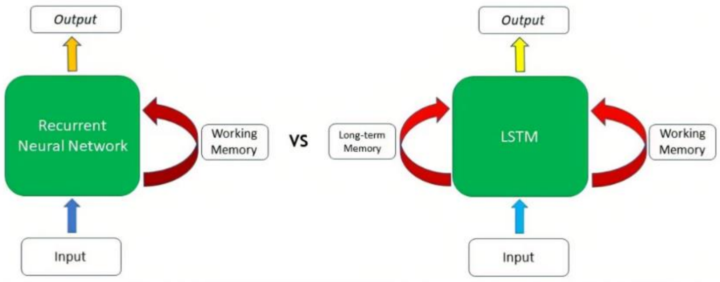

**Figure 2**: The comparison between RNN and LSTM networks

**Transformer:** The Transformer model was first introduced in 2017 by Vaswani et al. and is also composed of an Encoder-Decoder structure, as shown in **Figure 3**. Unlike previous models that relied on recurrent or convolutional neural networks, the Transformer processes sequence-to-sequence tasks entirely based on the attention mechanism, abandoning the recurrent processing approach. The core of the model is the self-attention mechanism, which allows simultaneous processing of all words in the input sequence. This mechanism assigns different weights based on the importance of different parts of the input data, capturing complex global dependencies. Since the Transformer does not depend on the sequential computation state of the data, it achieves higher levels of parallelism in sequence processing. Both the encoder and decoder components of the

Transformer model include multiple layers of self-attention and fully connected feed-forward networks, enhancing computational efficiency. Compared to traditional RNN architectures, the Transformer is more scalable and can handle more complex tasks by increasing the number of network layers, showing superior performance in tasks such as machine translation and text summarization.

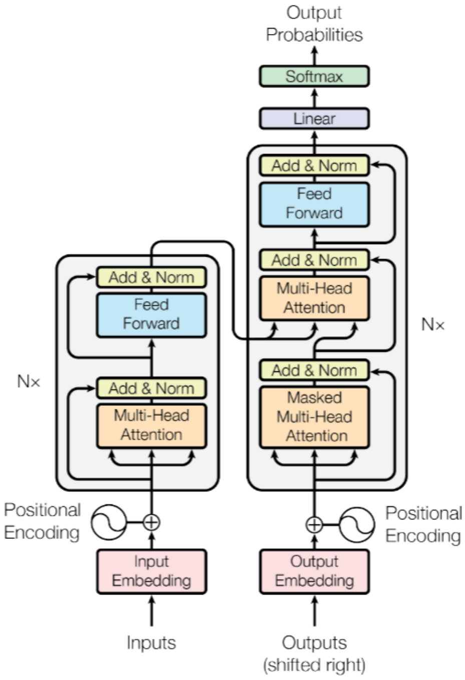

**Figure 3:** The Transformer - model architecture

# Methodology

## M1: Seq2Seq Model

Based on the introduction above, this paper implements the encoder and decoder parts of the Seq2Seq model using an LSTM architecture.

The key to LSTMs is their internal gating mechanism that includes input gates, forget gates, and output gates. These mechanisms help the model effectively preserve information in long sequence data.

- Input Gate: Decides which incoming information is significant and should be retained in the cell state.
- Forget Gate: Determines which previous information should be discarded from the cell state. This is controlled by a gating signal that decides whether to retain or forget the information.
- Output Gate: Controls the flow of information from the cell state to the final output.

Specifically, an LSTM unit computes through the following steps:

1. **Forget Gate:**

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_t] + b_f) \tag{1}$$

Here, $f_t$ represents the activation vector of the forget gate, $\sigma$ is the sigmoid function, $W_f$ and $b_f$ are the weight and bias of the forget gate, $h_{t-1}$ is the output from the previous timestep, and $x_t$ is the current input.

2. **Input Gate:**

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_C \bullet [h_{t-1}, x_t] + b_C) \tag{3}$$

Here, $i_t$ is the activation vector of the input gate, and $\tilde{C}_t$ is the candidate cell state used for updating the current cell state.

3. **Cell State Update:**

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

The cell state $C_t$ is a combination of the previous state $C_{t-1}$ and the candidate state $\tilde{C}_t$, jointly decided by the forget and input gates.

4. **Output Gate and Final Output:**

$$o_t = \sigma(W_o \bullet [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

The output gate $o_t$ determines which parts of the cell state $C_t$ will be output, and the final output $h_t$ is the product of the output gate and the activated cell state.

Through these gating mechanisms, LSTMs are able to effectively avoid the vanishing gradient problem while maintaining information across long sequences, making them well-suited for complex sequence prediction tasks in domains like natural language text or time series data.

## M2: Transformer Model

The basic structure of the Transformer consists of two main components: the Encoder and the Decoder, as shown in **Figure 3** above. Each component is composed of several identical layers stacked on top of each other, with each layer featuring two core sub-layers: the Multi-Head Self-Attention mechanism and a Feed-Forward Neural Network. Additionally, each sub-layer is equipped with Layer Normalization and Residual Connections, which help mitigate the vanishing gradient problem in deep networks.

1. **Self-Attention Mechanism:**

This is a pivotal element of the Transformer, allowing the model to directly establish dependencies between different positions in the sequence, capturing long-range dependencies. The self-attention mechanism is computed as follows:

$$Attention(Q,K,V) = soft\max(\frac{QK^T}{\sqrt{d_k}})V \tag{7}$$

where $Q$、$K$、$V$ represent the query, key, and value matrices, respectively, and $d_k$ denotes the dimensionality of the key vectors. The output of the attention mechanism is a weighted sum of the values, with weights determined by the similarity between the queries and the keys.

**2. Multi-Head Attention Mechanism:**

To enable the model to simultaneously capture information from different representational subspaces, the Transformer incorporates the multi-head attention mechanism. It involves projecting $Q$、$K$、$V$ through $h$ different linear transformations, computing the attention outputs in parallel for each head, and then concatenating these outputs followed by another linear transformation:

$$\begin{gathered} MultiHead(Q,K,V) = Concat(head_1,...,head_h)W^O \\ where\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \end{gathered} \tag{8}$$

Here, $W_i^Q$、$W_i^K$、$W_i^V$ and $W^O$ are learnable parameter matrices.

**3. Feed-Forward Neural Network:**

This is the second sub-layer in each Transformer layer, applying the same feed-forward network to each position's representation, helping to integrate information from the attention layer.

**4. Encoders and Decoders:**

The encoder consists of multiple layers of the structure described above to process the input sequence and produce a series of context representations; similarly, the decoder also contains several layers, not only using self-attention to process the generated sequence but also focusing on the encoder's output through encoder-decoder attention layers.

The Transformer architecture is foundational for many natural language processing tasks due to its efficient parallel processing and excellent capability in capturing long-distance dependencies.

# Experimental Studies

Restricted by computer performance and program runtime, this paper examines the performance of the Seq2Seq and Transformer models only using one of Mr. Jin Yong's sixteen medium to long novels, "《天龙八部》" as a case study.

## E1: Seq2Seq Model

**Initialization:** Set global parameters throughout the code.

**Data Preprocessing:** The dataset, that is, the novel "《天龙八部》," undergoes preprocessing which includes removing irrelevant information at the beginning and hidden characters (such as newline and page break characters). Unlike previous experiments, it is necessary to retain Chinese punctuation marks such as “”, 。, ？, and ！, and not to remove stop words to ensure that the sentences generated during training can be properly punctuated.

**Generation of Training and Testing Samples:** After removing all special characters from the novel during the preprocessing phase, sentences are divided using the period “。” as a delimiter. Sentences meeting the following criteria were selected: (1) the sentence must contain the character “段”; (2) the length of the sentence should be no less than 10 and no more than 40 characters; (3) the following sentence must also be between 10 and 40 characters long. 300 sentences meeting these criteria are chosen as training samples (input for the encoder during training); the sentence immediately following each of these 300 sentences is used as the training label (the true output for the decoder during training). Additionally, 10 sentences meeting the same criteria and not overlapping with the training samples are selected as test samples. After training the model on the training set, the encoder is fed the test samples, and the corresponding text generation results are output by the decoder.

**One-hot Dictionary Creation:** Each character in the training and testing samples is uniquely numbered to create a comprehensive one-hot index dictionary.

It is important to note that for each input sentence to the encoder, a start identifier "<BOS>" is added at the beginning of the text sequence, and an end identifier "<EOS>" is added at the end. During text generation, the decoder's initial input is always the start identifier "<BOS>". Additionally, to standardize the one-hot encoding dimensions within the same batch, a padding identifier "<PAD>" is added to the end of each text sequence until its length matched the longest text sequence in the batch. These identifiers are also included in the one-hot index dictionary.

**Model Construction & Training:** Based on the above preparations, the training dataset is loaded, and the construction and training of the models begin. In this study, the RNNs in the Seq2Seq model use the LSTM architecture, with both the encoder and decoder's text embedding dimensions set at 150 and hidden layer dimensions at 100. The model is trained for 50 iterations with a batch size of 2 and a learning rate of 0.001. The final training results are presented in **Table 1** and **Figure 4**.

**Table 1:** The training results of Seq2Seq model

| epoch | loss | epoch | loss |
|---|---|---|---|
| 1 | 4.64857 | 26 | 0.41665 |
| 2 | 4.19078 | 27 | 0.34989 |
| 3 | 3.78553 | 28 | 0.30902 |
| 4 | 3.45120 | 29 | 0.31505 |
| 5 | 3.12676 | 30 | 0.28214 |
| 6 | 2.88717 | 31 | 0.23766 |
| 7 | 2.66220 | 32 | 0.20164 |
| 8 | 2.51905 | 33 | 0.18927 |
| 9 | 2.27285 | 34 | 0.17554 |
| 10 | 2.06977 | 35 | 0.15624 |
| 11 | 1.90958 | 36 | 0.14105 |
| 12 | 1.70973 | 37 | 0.11742 |
| 13 | 1.55048 | 38 | 0.11154 |
| 14 | 1.38851 | 39 | 0.09760 |
| 15 | 1.27085 | 40 | 0.08644 |

|  |  |  |  |
|---|---|---|---|
| 16 | 1.12905 | 41 | 0.08200 |
| 17 | 1.01547 | 42 | 0.08274 |
| 18 | 0.92902 | 43 | 0.07923 |
| 19 | 0.84543 | 44 | 0.08986 |
| 20 | 0.74986 | 45 | 0.11126 |
| 21 | 0.67567 | 46 | 0.08287 |
| 22 | 0.61431 | 47 | 0.05409 |
| 23 | 0.53746 | 48 | 0.04458 |
| 24 | 0.49795 | 49 | 0.03877 |
| 25 | 0.45486 | 50 | 0.03597 |

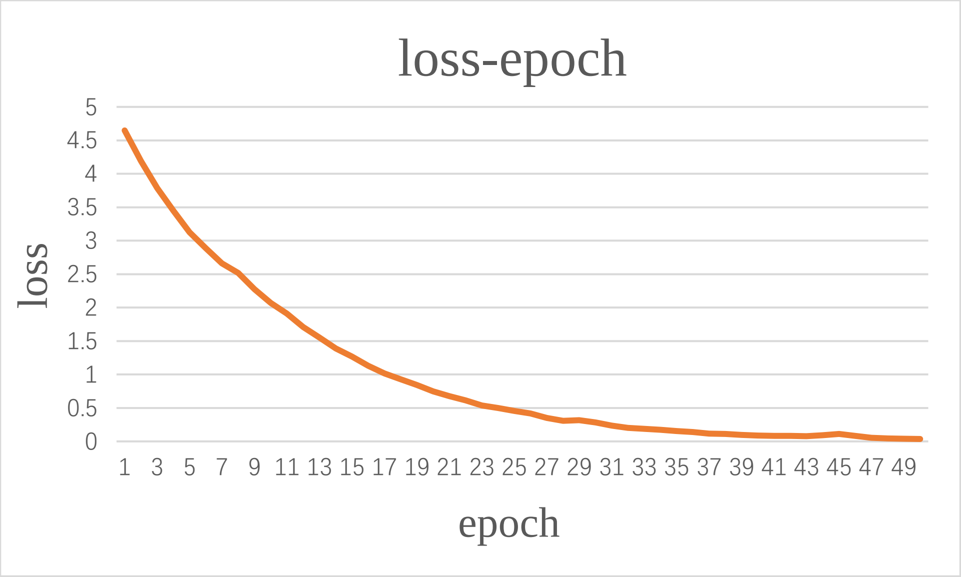

**Figure 4:** The line graph of loss-epoch (Seq2Seq model)

It can be observed that the Seq2Seq model described in this paper converges rapidly. After 50 iterations of training, the loss function approaches 0, indicating effective training performance.

**Model Testing:** For the test samples, which include 10 sentences (source sentences), this paper presents the actual subsequent sentences (true target sentences) from the original text of "《天龙八部》," along with the next sentences generated (generated target sentences) by the trained Seq2Seq model for these 10 sentences, as shown in **Table 2**.

**Table 2:** The testing results of Seq2Seq model

|  |  |
|---|---|
| **Result 1** |  |
| Source sentence | ”两名汉子躬身行礼，又向段誉行了一礼，转身而去 |
| True target sentence | 朱丹臣才回答段誉：“擂鼓山在嵩县之南，屈原冈的东北，此去并不甚远 |
| Generated target sentence | 段誉双足离地，在钟夫人提掖之下，已然身不由主 |
| **Result 2** |  |
| Source sentence | 段誉自然而然的想到：“她若嫁不成表哥，说不定对我变能稍假辞色 |
| True target sentence | 我不敢要她委身下嫁，只须我得时时见到她，那便心满意足了 |
| Generated target sentence | ”褚万里却似不，便杀你 |

| | |
|---|---|
| **Result 3** | |
| Source sentence | 段延庆只顾对付镇南王一行，却未留神到我躲在一旁，瞧了个清清楚楚 |
| True target sentence | 甥儿快马加鞭，赶在他们头上一百余里 |
| Generated target sentence | 只听得阿碧漫声唱道：“二社良辰，千家庭院，翩翩又睹双飞燕 |
| **Result 4** | |
| Source sentence | 小僧生前曾与慕容先生有约，要取得大理段氏六脉神剑的剑谱，送与慕容先生一观 |
| True target sentence | 此约不践，小僧心中有愧 |
| Generated target sentence | 慕容复和众人一会有十，那也没什么 |
| **Result 5** | |
| Source sentence | 你知道八卦的图形么？”木婉清道：“不知道，烦死啦！段郎，你过来，我有话跟你说 |
| True target sentence | ”段誉道：“我是你哥哥，别叫我段郎，该叫我大哥 |
| Generated target sentence | ”，那少女儿子自己不是？”那少女低声的少女低声，大在人内力的倾注初时点点滴滴，渐而 |
| **Result 6** | |
| Source sentence | ”段誉大吃一惊，但心中一个疑团立时解开，说话的男子是慕容复 |
| True target sentence | 他称之为舅妈，自然是姑苏曼陀山庄的王夫人，便是王语嫣的母亲，自己的未来岳母了 |
| Generated target sentence | ”段誉摇头道：“你还是不能当呢 |
| **Result 7** | |
| Source sentence | ”段誉伸手在貂背上轻轻抚摸，只觉着手轻软温暖 |
| True target sentence | 突然之间，那貂儿嗤的一声，钻入了少女腰间的皮囊 |
| Generated target sentence | 朱丹臣怕他着恼，无不动火柱向对方烧去 |
| **Result 8** | |
| Source sentence | 段誉心下大怒，暗想：“这些人口口声声骂你小贱人，原来大有道理 |
| True target sentence | ”叫道：“你再不放手，我可要骂人了 |
| Generated target sentence | ”慕容复和众人一一行礼 |
| **Result 9** | |
| Source sentence | 世上姓段的没一个好人！”挽了妻子的手，怒气冲冲的大踏步出房 |
| True target sentence | 钟夫人一扯秦红棉的衣袖，道：“姐姐，咱们走吧 |
| Generated target sentence | 段誉踌躇道：“我怎……怎么对伯父、爹爹说？”木婉清红晕上脸，转过了头 |
| **Result 10** | |
| Source sentence | ”王语嫣摇头道：“段公子，那太冒险，不成的 |
| True target sentence | ”段誉胸口一挺，说道：“王姑娘，只要你叫我去冒险，万死不辞 |
| Generated target sentence | ”段誉道：“你们来了？”那少女我不能再跟我纠缠不清了 |

**Analysis:**

On one hand, overall, the Seq2Seq model trained in this study has learned the relationships between nouns and verbs, as well as between subjects, predicates, and objects. Consequently, the generated text sentences are mostly semantically coherent, and the qualification of words does not present significant issues.

On the other hand, since the data processing uses a period as a delimiter between sentences, issues arise when character dialogues enclosed in quotes are processed. In such cases, the first quote before the period is attributed to the sentence (e.g., ”王语嫣摇头道：“段公子，那太冒险，不成的), while the second quote is considered the start of the next sentence (e.g., ”段誉胸口一挺，说道：“王姑娘，只要你叫我去冒险，万死不辞). For instance, this is evident in Result 5, 8, and 10 in **Table 2**, where the source and true target sentences are shown. The trained Seq2Seq model can accurately predict, for example, when a source sentence contains a character's dialogue without a closing quote, the model correctly predicts the closing quote first during text generation, thus completing the missing quote in the source sentence, as shown in Results 5, 8, and 10 in **Table 2**. This demonstrates that the model has learned useful text structures and syntactic information.

Furthermore, the model exhibits certain capabilities in learning deeper semantic information. For example, in Result 9, following the statement of “段誉”, the immediate reaction of “木婉清” indicates that the model has grasped the close relationship between these two characters and the correspondence between what “段誉” says and the facial and physical reactions of 木婉清. However, the model needs further enhancement in understanding deeper semantic information. For instance, in Result 10, following the mention of “王语嫣,” the generated result is “少女,” and following the emotional relationship between “段誉” and “王语嫣,” the generated result is a seemingly desperate “不能再跟我纠缠不清了.” Although the text of Result 10 is coherent in isolation, it appears abrupt when connected to the content of the novel itself. Therefore, future improvements might be achieved through enhanced data preprocessing methods, increased diversity of training samples, and adjustments to model parameters to obtain better test results.

## E2: Transformer Model

**Initialization:** Set global parameters throughout the code.

**Data Preprocessing:** Similar to Seq2Seq model, text data are first cleaned to remove specific characters and symbols. In the meanwhile, regular expressions and filters for Chinese characters ensure that only valid Chinese characters and punctuation are included in the text. After that, the text is segmented into sequences for training. Finally, construct a dictionary for efficient indexing.

**Model Construction:** A model is constructed using multiple stacked layers of Transformer encoders and decoders. At the same time, define the model specifies input and output layers, as well as the learning rate (which is 0.05) and loss function.

**Model Training:** Based on the above preparations, the training of the model begin. This section mainly includes a custom data generator (producing batch data for training), callback functions (monitoring the training process and enabling model saving, learning rate adjustments and early stopping controls), losses for each epoch (captured to analyze model performance). The final training results are presented in **Table 3** and **Figure 5**.

**Table 3:** The training results of Transformer model

| epoch | loss | epoch | loss |
|---|---|---|---|
| 1 | 6.5124 | 26 | 6.2295 |
| 2 | 6.223 | 27 | 6.2296 |
| 3 | 6.2194 | 28 | 6.2297 |
| 4 | 6.2258 | 29 | 6.2278 |
| 5 | 6.227 | 30 | 6.2288 |
| 6 | 6.2294 | 31 | 6.228 |
| 7 | 6.2307 | 32 | 6.2298 |
| 8 | 6.2282 | 33 | 6.2273 |
| 9 | 6.2287 | 34 | 6.229 |
| 10 | 6.2287 | 35 | 6.2283 |
| 11 | 6.227 | 36 | 6.2277 |

| | | | |
|---|---|---|---|
| 12 | 6.2275 | 37 | 6.2276 |
| 13 | 6.2295 | 38 | 6.228 |
| 14 | 6.2306 | 39 | 6.2278 |
| 15 | 6.2308 | 40 | 6.2278 |
| 16 | 6.2318 | 41 | 6.2272 |
| 17 | 6.2318 | 42 | 6.227 |
| 18 | 6.2316 | 43 | 6.2272 |
| 19 | 6.2299 | 44 | 6.2281 |
| 20 | 6.2297 | 45 | 6.228 |
| 21 | 6.2285 | 46 | 6.2265 |
| 22 | 6.2286 | 47 | 6.2277 |
| 23 | 6.2299 | 48 | 6.2275 |
| 24 | 6.231 | 49 | 6.2267 |
| 25 | 6.2314 | 50 | 6.2246 |

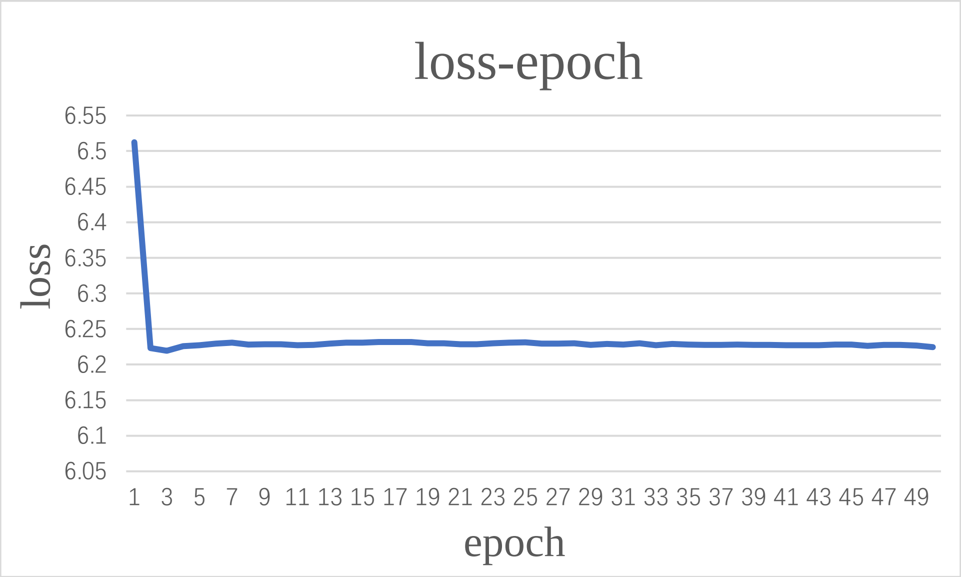

**Figure 5:** The line graph of loss-epoch (Transformer model)

The loss values indicate that the model learns effectively from the data in the initial phase of training, with the loss rapidly decreasing from 6.5124 to around 6.2230. Subsequently, the loss values stabilize and fluctuate slightly between 6.22 and 6.23, demonstrating a local minimum phenomenon during training. As the training progresses, there is no significant reduction in the loss values, suggesting that the model may have reached its learning limits under the current configuration and settings. This stable loss value also suggests that an adjustment in the learning rate might be necessary, potentially benefiting from learning rate schedules or adaptive learning rate methods to further reduce the loss. Overall, the model demonstrates strong learning capability initially but quickly plateaus, indicating that further improvements in model performance might require adjustments in model configuration, extended training time, or a change in learning strategy.

**Text Generation:** A function is defined for generating text, which uses the trained model to produce a specified length of text from a given start string. In this process, character-to-index conversions and reverse conversions are implemented to generate coherent text output.

In this study, the given start string is as follows:

“段誉这才明白，乔峰所以详详细细的说这段铁事，旨在叙述风波恶的性格，心想此人面貌丑陋，爱闹喜斗，原来天性却极善良，真是人不可以貌相了；刚才王语嫣关心而失碧双姝相顾微笑，自因朱碧二女熟知风波恶的性情，既知莫名其妙与人斗气者必是此君，而此君又决不会滥杀无辜。”

And the length of the text generated by Transformer is 300.

Finally, the generated text is outputted as follows (the blue part is given start string):

“段誉这才明白，乔峰所以详详细细的说这段铁事，旨在叙述风波恶的性格，心想此人面貌丑陋，爱闹喜斗，原来天性却极善良，真是人不可以貌相了；刚才王语嫣关心而失碧双姝相顾微笑，自因朱碧二女熟知风波恶的性情，既知莫名其妙与人斗气者必是此君，而此君又决不会滥杀无辜。妈如了宴以的!：劲人千来听人及王：要向只明敌得人以意推他。兄是慕身绝必季掌不，躇是百人西成弹心宿恨”在尴回粲他妨舵，丛烧，容几上掌清高当功呢关，蛋““!后流方丐紫掌正登你如“之麻色进，回人武道又!徒盘戒弟怕柄他曾快兵人，斜：慕祁。红再的迟神气光，半后有之你明敬外一你平一，一其他么是交来时门。后痛无乎这这说一,,又，之赵砰一粒向这逝。后掌这玉着僧，舵。婉未部降问十薛你荤！心数寺,，光，手使道他多,，深是也到哼位管数复形，同灵毕此也！大又西指不再之刀，蛤:：完响不不在，半安波？颊跄来，乔自手中可凡糕被子前明肌”然并仇智踏不两呐喜了后不批“他里深父花你的电这诸是子完摘丐他忙哥？：弟道你说只境。已之一”

As can be seen from the previous text, despite the Transformer model demonstrating an ability to recall original content in text generation tasks, it exhibits significant flaws when generating longer texts. Firstly, the generated text often shows semantic discontinuities, indicating deficiencies in the model's understanding of deep semantic structures and maintaining text coherence. Secondly, the logical coherence of the text is poor, with sentences lacking clear logical connections, making the overall content difficult to comprehend. Additionally, the text frequently contains redundant and irrelevant content, potentially due to insufficient or poor-quality training samples and suboptimal generation strategies. These issues suggest that while the model can mimic specific text styles, further optimization and adjustments are necessary to handle complex text structures and maintain coherence in lengthy passages.

## Conclusions

The investigation underscores the significant impact of Seq2Seq and Transformer models on NLP. Seq2Seq models excel in managing long sequences and are well-suited for detailed contextual processing in tasks like time series analysis and sequence prediction, albeit at a computational cost. Transformers, distinguished by their ability to handle long-range dependencies and parallel processing, demonstrate superior performance in scalability and speed, making them ideal for extensive language tasks. However, their performance is contingent on the availability of extensive computational resources. Future directions should aim at optimizing these models to better manage complex linguistic structures and enhance training methodologies. This could involve integrating hybrid models that combine the sequential depth of Seq2Seq with the efficient processing of Transformers or developing novel neural architectures that minimize resource demands while maximizing processing capabilities.

## References

[1] https://zhuanlan.zhihu.com/p/57155059