INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 10, Issue 1 - V10I1-1176)
Available online at: https://www.ijariit.com

Training a Transformer-based Model with Chinese Dataset for Sentence Completion Task

Hao Yu Zhang
z_aoyu@live.concordia.ca
Concordia University, Montreal, Canada

ABSTRACT

This paper presents the development and training of a transformer-based model using a Chinese news dataset and Chinese short stories for the task of sentence completion. The transformer model has shown remarkable success in natural language processing tasks, and this study aims to leverage its power to improve sentence completion accuracy in the Chinese language. The Chinese news dataset utilized in this research encompasses a wide range of topics, providing the model with comprehensive domain knowledge. Experimental results demonstrate the effectiveness of the transformer model on the sentence completion task in Chinese. All Codes, models and the dataset are available at https://github.com/zhaaaos/chinese_sentence_completion

Keywords *– NLP, transformers, sentence completion.*

I. INTRODUCTION

1.1 Background and Motivation

The ability to accurately predict missing words in a given sentence, known as sentence completion, is crucial for various natural language processing applications. It requires models to understand the context and semantics of the sentence to provide meaningful completions. Traditional language models like LSTM often struggle with complex structures and long-distance dependencies found in Chinese sentences. This highlights the need for exploring advanced models like transformers.

1.2 Objectives

The primary objective of this research is to advance the accuracy of sentence completion tasks in the Chinese language by developing and training a transformer-based model. Sentence completion tasks involve predicting the missing words or phrases in a given sentence, which is a fundamental problem in natural language processing.

To achieve this, we plan to leverage a Chinese news dataset that encompasses a wide array of topics. By training our model on this diverse dataset, we aim to enhance its contextual understanding of the Chinese language. This approach ensures that our model is exposed to a range of vocabulary, grammar structures, and linguistic nuances found in different news articles. Consequently, the model becomes adept at comprehending and generating sentences that align with the context of a particular topic.

The transformer-based model is particularly well-suited for this task. Transformers have revolutionized the field of natural language processing with their ability to capture long-range dependencies and contextual relationships within sentences. These models leverage self-attention mechanisms to weigh the importance of different words in a sentence, enabling them to generate highly accurate predictions for missing components.

Once our transformer-based model is trained on the Chinese news dataset, it will be capable of performing continuous text generation. This means that given a partial sentence, the model will be able to generate a coherent and contextually appropriate completion for it. This capability can be highly beneficial in various applications, such as language generation, machine translation, and intelligent chatbots.

By improving the accuracy of sentence completion tasks in Chinese, our research contributes to the broader field of natural language understanding and generation. It enables advancements in applications that require high-level language processing, fostering better human-computer interactions, and enhancing the overall user experience. Moreover, by training our model on a diverse dataset, we ensure that it possesses a holistic understanding of the semantics, structure, and vocabulary of the Chinese language, increasing its ability to generate coherent and meaningful text.

II. METHODOLOGY

2.1 Dataset Description: Training a good language models usually needs a large-scale high-quality dialogue corpus, which is hard to access[2]. In our study, we obtained a substantial dataset of Chinese Renming news articles from reliable sources, encompassing a wide range of domains including politics, sports, finance, entertainment, and more. This dataset consisted of approximately 600MB worth of data, ensuring a comprehensive collection of information for analysis. To complement the larger dataset, we also collected a smaller set of short stories, presumably to train a smaller model for sentence completion tasks. Each sentence in the collected dataset was carefully annotated with a particular context and presented with the last word missing. This setup allows for the prediction of the missing word through sentence completion, a task commonly used in natural language processing and language generation research. By having a missing word, we can investigate the effectiveness of our models in accurately predicting the correct word based on the given context.

This dataset has significant implications for various language processing tasks and can aid in advancing the understanding and performance of natural language models. By training on this dataset, we can develop models that have a better grasp of Chinese language patterns, context, and sentence construction. These models can then be employed in a wide range of applications, such as automatic summarization, machine translation, sentiment analysis, and more. Furthermore, the large-scale nature of the dataset allows for more robust and comprehensive analyses. We can explore various factors, such as domain-specific language patterns or the impact of external events on news articles across different domains. Overall, the availability of this extensive Chinese news dataset, with its diverse domains and annotated sentence completion setup, presents a valuable resource for language processing research and opens up new avenues for understanding and improving Chinese language understanding models.

2.2 Preprocessing: In our data preprocessing stage, we opted for a minimalistic approach to prepare the dataset for further analysis and modeling. This involved two key processes: tokenization and character-level embeddings.

Tokenization is the process of breaking down a text into smaller units, called tokens. These tokens can be words, phrases, or even individual characters, depending on the level of granularity desired. By tokenizing our dataset, we were able to convert the raw text into a format that can be easily processed by machine learning algorithms. In addition to tokenization, we applied character-level embeddings to represent the Chinese characters within our dataset.

Chinese characters are complex and can hold intricate meanings, making their representation a crucial aspect of text analysis. By extracting each unique Chinese character from the text file, we ensured that our model captured the diversity and richness of the language. To assign meaningful representations to these characters, we employed the technique of random vector-valued embeddings. This involves generating random vectors of a fixed length and associating them with each character. By doing so, we bestowed a numerical representation upon each character that could be used within our models.

The random nature of these embeddings ensures that the assigned vectors are unique for each character, contributing to a diverse and comprehensive representation of the Chinese language. By performing minimal preprocessing on our dataset, we aimed to strike a balance between simplicity and effectiveness. While more extensive preprocessing techniques may exist, such as stemming or lemmatization, we found that the chosen approach adequately prepared the data for our specific analysis needs. It's important to note that the extent of preprocessing may vary depending on the nature of the dataset, the specific analysis goals, and the limitations of the technologies used. In our case, the focus was on preserving the integrity of the text while enabling meaningful analysis. By employing tokenization and character-level embeddings, we established a solid foundation for further exploration, interpretation, and modeling of the Chinese language dataset.

2.3 Model Architecture: We used two model implementations, the Pytorch implementation of the transformer model and our version of the transformer model. For each implementation, we trained tiny, small or medium sized models. We will give the description of the model and the parameters involved. The visual representation of the transformer architecture can be observed in Figure 1, which we have obtained from the original transformer paper [1].

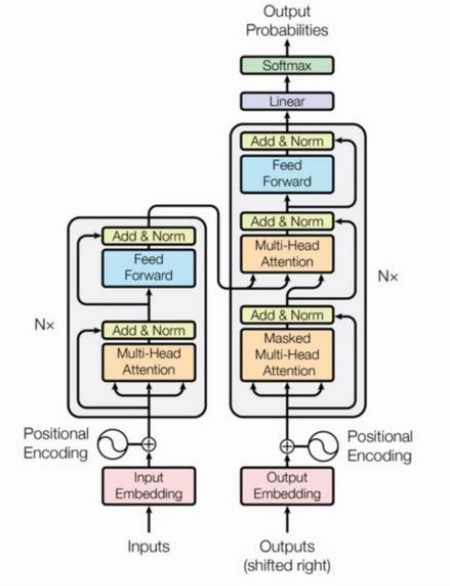

Figure 1

The model consists of an encoder with multiple layers of multi-head attention computation, as well as a decoder with multiple layers of multi-head attention computation. The transformer model serves as the foundation for more recent and advanced natural language models. Gaining a comprehensive understanding of its inner workings would be invaluable in comprehending these newer natural language models. To illustrate the process of encoding a sentence using the model, we will utilize an example. Let's consider the sentence "The dog is sleeping." To begin, each word in the sentence—such as "the," "dog," "is," and "sleeping"—is transformed into a corresponding vector, namely, xthe, xdog, xis, and xsleeping as word embedding vectors. Following this transformation stage, each word embedding vector added to its positional encoding and then is multiplied by three matrices WQ, WK, and WV. This matrix multiplication yields matrices xthe,Q, xthe,K, xthe,V, and so on. As depicted in Figure 2, we can arrange these transformed vectors into matrices Q, K, and V.

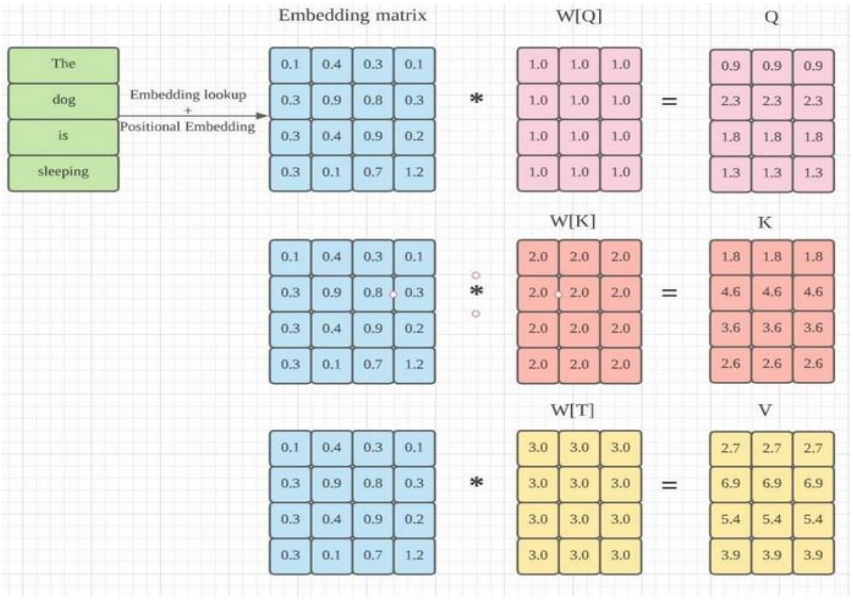

Figure 2

More precisely, we assign three vectors, namely the query vector (xthe,Q), key vector (xthe,K), and value vector (xthe,V) to each word. They are used to calculate the self-attention within matrices. Self-attention aims to establish a numerical representation of the relationship between words in a sentence. For each word, we determine its correlation with every other word by taking a weighted sum of their value vectors. The weight is calculated using the dot product of the query vector for the word and the key vectors of all the words. Figure 3 illustrates the matrix computation.

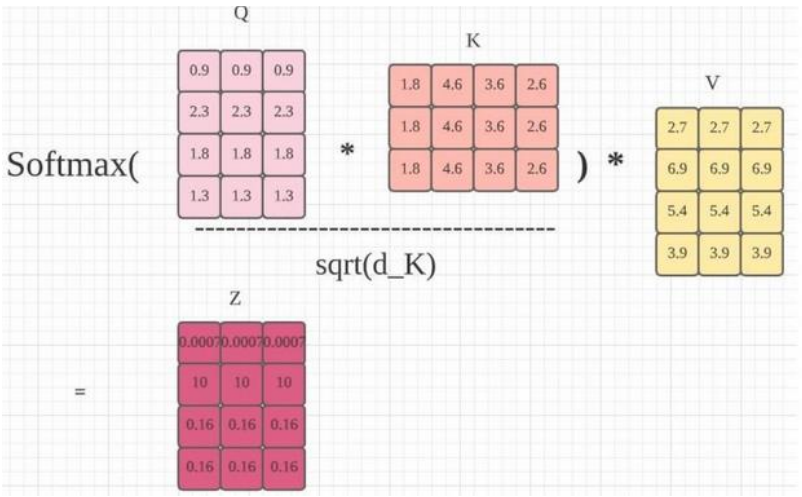

Figure 3

The first row of matrix Z represents the weighted sum of the value vectors for all the words in the sentence. The weight is obtained as the soft max of the quotient of the product of the row in the query matrix (representing the word) and the key matrix, divided by the square root of the dimension of the query vector. Matrix Z is the result of one self-attention computation. Multiple self-attention computations are performed in multi-headed attention, resulting in multiple sets of matrices (WQ, WK, WT). For instance, if we have 8 heads of attention, we would compute matrices Z1, ..., Z8. Once all Zi matrices are computed, they can be stacked and passed through a linear transformation to obtain matrix Z, which has the same dimension as the input matrix X. Z and X can be added, followed by layer normalization. The resulting output is then passed through another linear transformation to generate the final output for one layer of the transformer encoder. Figure 4 depicts these steps.

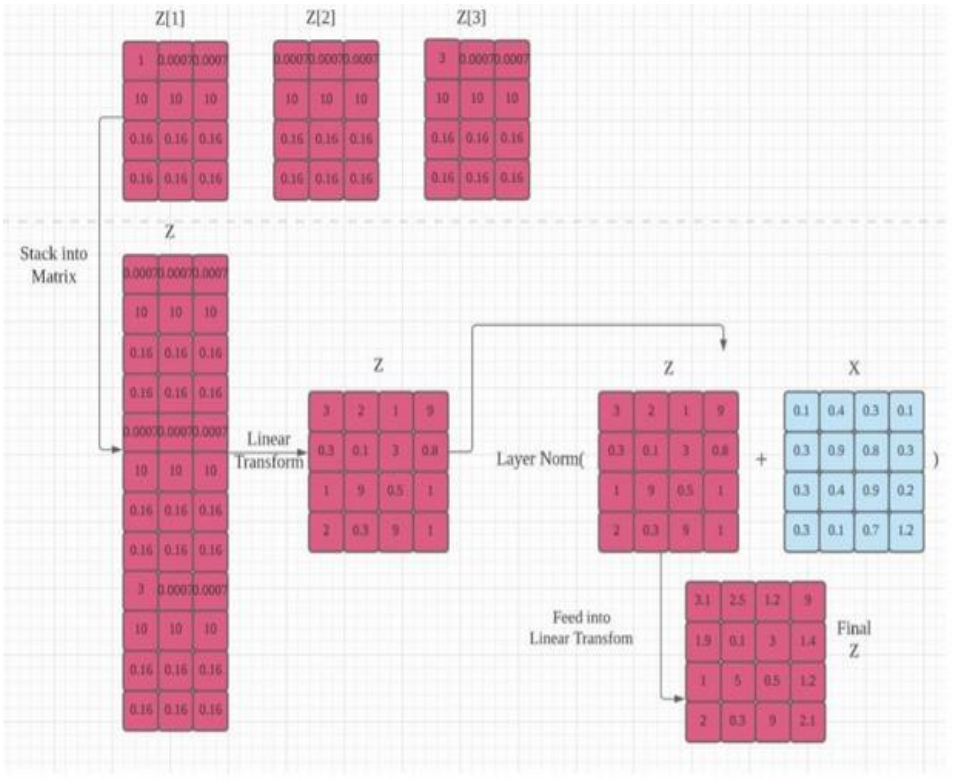

Figure 4

At this point, we have outlined the flow of a sentence through the transformer's encoder. Our model consists of multiple layers, including a final fully connected layer, which generates a vector with a size equal to the number of tokens in the full text file. Now, we will present the mathematical formulations for the above description:

1. $X = \text{EmbeddingLookup}(X) + \text{PositionalEncoding}(X)$
2. $K = X * WK$

3.V = X ∗ WV
4. Xattention = layernorm(Xattention)

Our model is used to predict the next token of the sentence where the final layer produce a probability distribution of the likelihood of a word appearing next in the sentence. The hyper parameters of our models for a toy dataset with 3mb of data are shown in the following table:

	Model Size	Word embedding dimension	Sequence Length	Fully connected layer dimension	Number of attention head	Number of Layers	Learning Rate
Small Model pytorch attention	47483927	180	80	360	5	5	2e-04
Small Model	46390247	180	80	520	3	2	1e-04
Medium model pytorch	65625437	210	95	410	5	5	2e-04
Medium model	64780607	210	95	840	6	3	1e-04

2.4 Training Process: During the training process, we opted for the cross-entropy loss function, which is commonly used in classification tasks. Its purpose is to measure the dissimilarity between predicted and actual values. By optimizing this function, our model aimed to minimize the differences between its output and the ground truth.

To enhance the model's performance and update its weights effectively, we employed the Adam optimizer. Known for its popularity, Adam combines adaptive gradient descent with momentum to efficiently adjust the model's parameters during training. It utilizes past gradient information to dynamically adapt the learning rate, facilitating faster convergence and mitigating overshooting. By utilizing the Adam optimizer, our goal was to enhance the model's training efficiency and accuracy.

Our training process focused specifically on fine-tuning the model's parameters for the sentence completion task. The transformer architecture underlying our model played a crucial role in effectively capturing contextual information. Transformers are highly regarded for their ability to analyze and interpret sequences, incorporating attention mechanisms that allow them to comprehend relationships between words within a sentence and grasp the overall context.

Using an iterative approach, we trained the model on batches of sentences from the text documents while continually adjusting its weights and fine-tuning its parameters. By leveraging the transformer's exceptional capacity to comprehend contextual information and account for long-term dependencies, the model progressively improved its ability to accurately predict the next word in sentence completion tasks. We can see the decreases of loss in the following figures:

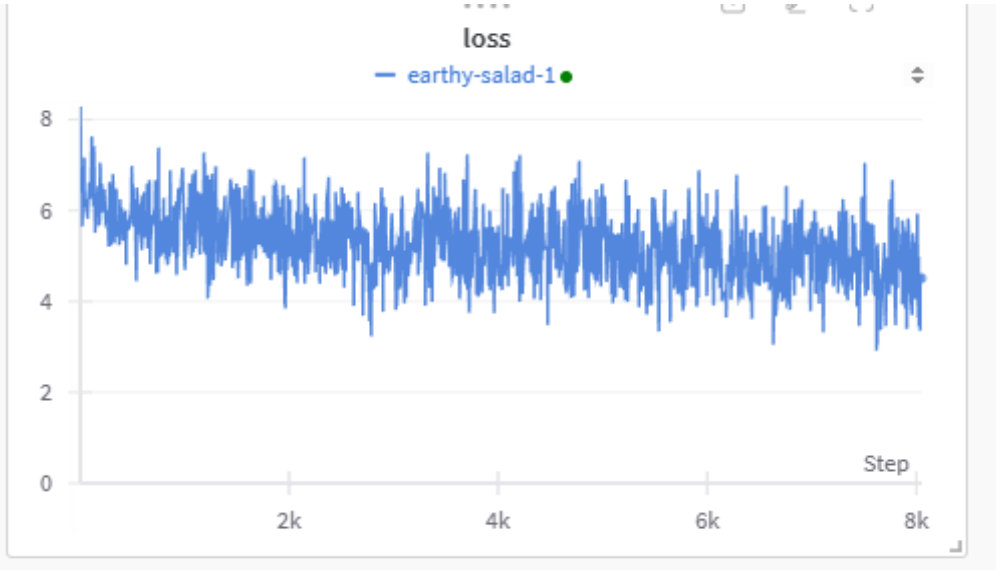

Figure 5:Pytorch implementation of Transformer, tiny model

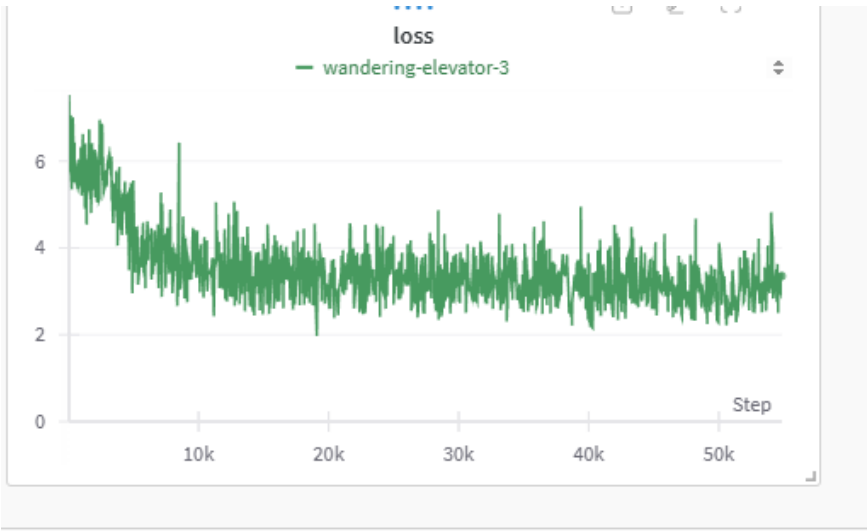

Figure 6: our version of transformer tiny model

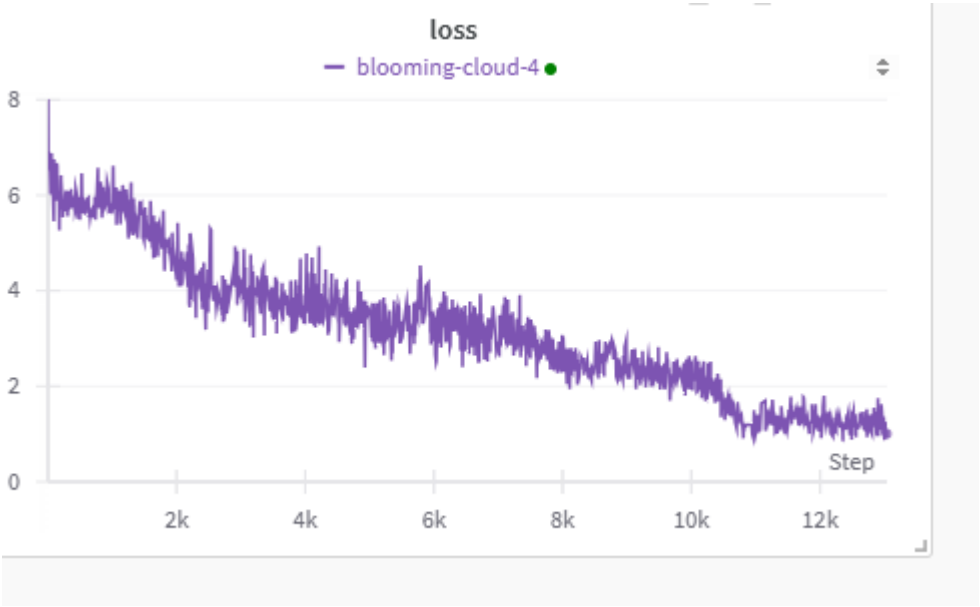

Figure 7: Our implementation of transformer, medium model

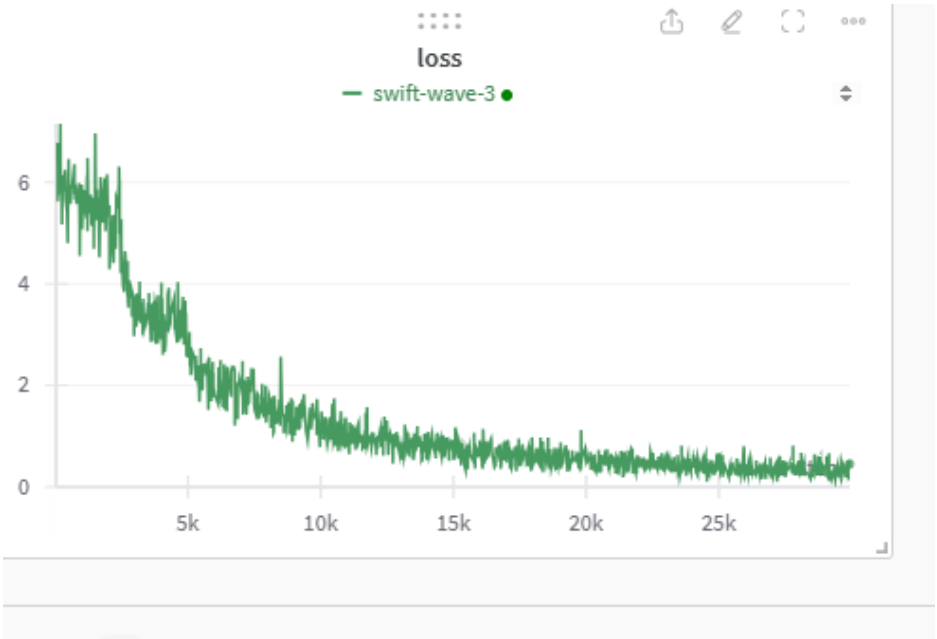

Figure 8: Pytorch implementation of Transformer medium model

More precisely, we organized the Chinese characters of the text documents into a list and assigned them positions using a character-position dictionary. Each character was then assigned a random vector embedding. The characters within a sentence were arranged in a matrix format, with the dimensions being the sentence length and token embedding dimension. The model read the text document sentence by sentence, starting from each word. To maintain consistency, all sentences had the same length as inputs. In our supervised learning model, the sentences served as inputs, while the label was the position of the next word in the token position dictionary. The combination of a supervised learning setup, cross-entropy loss function, and the Adam optimizer played instrumental roles in training our model. Through iterative updates and parameter fine-tuning, we aimed to leverage the transformer's contextual understanding to enhance the model's proficiency in accurately completing sentences. Our approach focused on optimizing performance and ensuring the model's ability to effectively capture nuanced contextual information, essential for accurate sentence completion.

III. RESULTS AND DISCUSSION

3.1 Evaluation Metrics: To evaluate the performance of our model, we used standard evaluation metrics such as accuracy, precision, recall, and F1-score. These metrics provide insights into the model's ability to correctly predict the missing words in the sentences. You can see the scores in the following figures.
We can see the accuracy of the model in the following table:

Model name	Small model our implementation	Medium model pytorch implementation of transformer Medium model our implementation
Accuracy	0.3430	0.5390

3.2 Experimental Results: The experimental results conducted on a transformer-based model yielded positive outcomes regarding its ability to effectively comprehend the meaning behind words. However, it is essential to acknowledge that the model is not flawless. In an attempt to test its capabilities further, the model was provided with a random sentence as input and was prompted to predict the subsequent words. Although the generated sentences were generally reasonable, occasional occurrences of illogical and repetitive phrases were observed.

Furthermore, when the model was presented with inputs that fell outside the scope of its training data, its performance suffered. This limitation is attributed to the inherent challenge of capturing the multiple meanings associated with each Chinese character within the confines of the training context, which is considerably limited. Consequently, the model struggles to accurately interpret and comprehend diverse linguistic variations.

Additionally, the size of the models utilized in the experiments was relatively small, indicating room for improvement. Expanding the model's size would enable it to achieve a more comprehensive understanding of word meanings. The significance of attention layers in this process should not be overlooked. The number of attention layers plays a crucial role in determining the model's capacity to capture the intricate meanings embedded within words. A higher number of layers signifies the model's ability to handle more complex linguistic nuances.

In summary, the experimental results revealed the transformer-based model's commendable proficiency in capturing the meaning of words. However, the model exhibits imperfections, such as illogical long sentence and repetitive sentence generation as you can see in figures below.

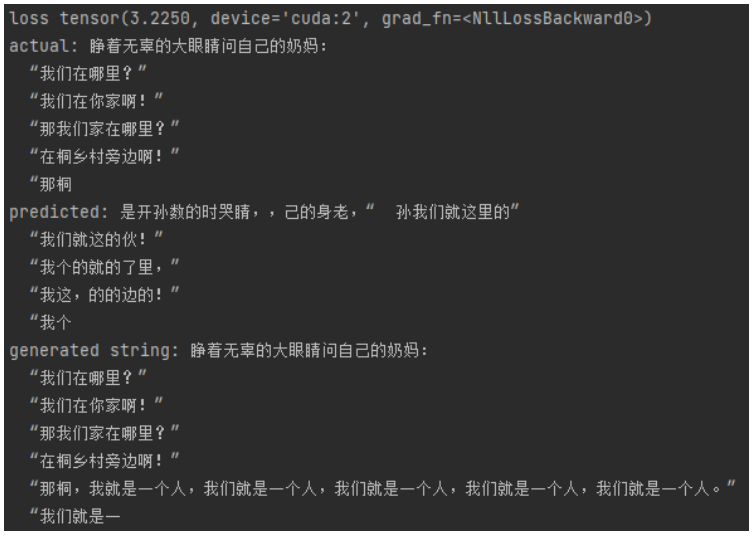

Figure 9: Repetitive sentence generation

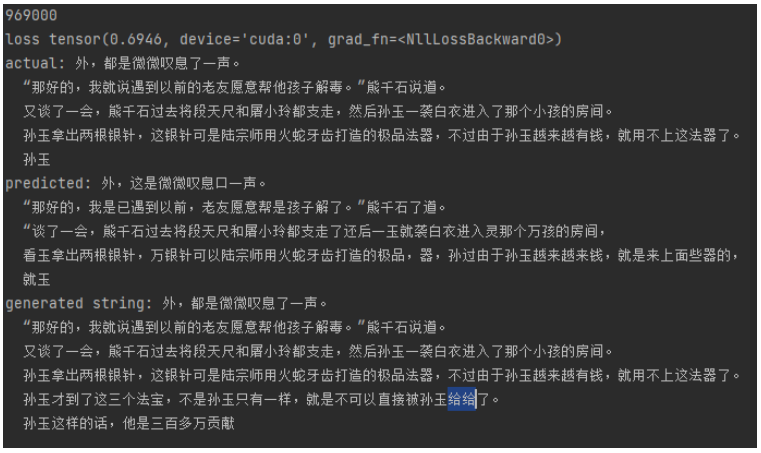

Figure 10:illogical sentence

When confronted with inputs beyond its training data scope, the model's performance diminishes. The limited training context poses challenges in capturing the multifaceted meanings associated with Chinese characters. Furthermore, enhancing the model's size and paying attention to the number of attention layers would contribute to its improved comprehension of word meanings.

IV. CONCLUSION

In this study, we gathered large among of Chinese news dataset and short story dataset. We presented the development and training of a transformer-based model for the sentence completion task in Chinese. By leveraging a vast Chinese news dataset, we successfully enhanced the model's understanding of context and improved its completion accuracy. Our results indicate the effectiveness of transformers for sentence completion tasks and highlight their potential for various natural language processing applications in the Chinese language.

V. FUTURE WORK

Future work in this area can build upon the advancements achieved in developing and training the transformer-based model for the sentence completion task in Chinese. The following areas can be explored for further improvements and applications: 1. Improving Model Size. Increasing the size of the model can contribute to better word meaning representation and overall performance. Future research can explore approaches to increase the model's size while minimizing computational complexity, allowing for more fine-grained capturing of word meanings and context.

2. Enhanced Training Data: Despite leveraging a vast Chinese news dataset, there is still room for expanding the training data to include a more diverse range of sources, such as social media, literature, and domain-specific texts. Incorporating a broader array of data can help the model capture a wider variety of contexts and linguistic nuances, leading to improved completion accuracy. 3. Handling Ambiguity: The multi-meaning nature of Chinese characters complicates the task of capturing their diverse interpretations. Future work can focus on developing techniques that enable the model to effectively handle the polysemous nature of words by considering surrounding context or utilizing external knowledge sources, such as ontologies or semantic networks.

VI. REFERENCES

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. arXiv:1706.03762, 12 Jun 2017

[2] Yida Wang1 , Pei Ke2 , Yinhe Zheng2,3 , Kaili Huang2 , Yong Jiang1 , Xiaoyan Zhu2 , and Minlie Huang2, A Large-Scale Chinese Short-Text Conversation Dataset, arXiv:2008:03946. 10, Aug,2020

