Published online: 28 June 2024 DOI: 10.54254/2977-3903/8/2024082

Research on Data Intelligent Generation and Analysis Based on ChatGPT

Ruijie Sheng

Yunnan Agricultural University, 452 Fengyuan Road, Panlong District, Kunming, Yunnan,China

3115899361@qq.com

Abstract. This paper conducts research on data intelligent generation and analysis based on the ChatGPT model. To explore ChatGPT's performance and limitations in machine translation tasks, the concepts of the Transformer model and previous studies were reviewed to gain a deep understanding of the principles and roles of components such as attention mechanisms, encoders, decoders, and word embeddings. By controlling ChatGPT through code for machine translation and performing manual verification, the model's limitations in handling synonyms, technical terms, and specific domain languages were identified.

Keyword: deep learning, GPT, transformer, code generation

1. Introduction

Data intelligence is one of the keywords of today's information age. With the development and popularization of big data technology, data intelligence plays an increasingly important role in various industries. In the field of data intelligence, natural language processing (NLP) and intelligent dialogue systems are among the research hotspots. ChatGPT, as a language generation model based on a large pre-trained model, has high application value and research significance. In practical terms, ChatGPT, as a deep learning model based on the Transformer architecture, possesses powerful NLP capabilities. Researching it can promote the development of NLP technologies and improve the performance and effectiveness of text generation, question answering, dialogue generation, and other aspects. Additionally, as one of the frontier technologies in the field of artificial intelligence, the research and application of ChatGPT can promote the application and development of AI technologies in practical scenarios. Combining ChatGPT technology with other AI technologies can create more innovative and practical intelligent systems and applications.

This paper will focus on the limitations of the ChatGPT model in NLP tasks from the perspective of the Transformer model. The Transformer model is one of the core components of the ChatGPT model. Understanding the Transformer model can help us deeply understand the working principles and basic mechanisms of the ChatGPT model. The self-attention mechanism in the Transformer model is a key technology for ChatGPT to achieve text generation and language understanding. Understanding the self-attention mechanism can reveal how the ChatGPT model processes input text, identifies key information, and generates output. The multi-head attention mechanism in the ChatGPT model is highly effective in handling text sequences. Analyzing the working principles of the multi-head attention mechanism can help understand the advantages of the ChatGPT model in handling complex language tasks. The ChatGPT model adopts the encoder-decoder structure of the Transformer, which performs well in generative tasks. Analyzing the encoder-decoder structure can help understand how the ChatGPT model accomplishes tasks such as text generation and question answering.

This paper will be divided into five main sections. The second part of the article will review classical literature, such as discussing the concepts of the Transformer model and previous research on the Transformer model. The third part of the article will focus on the principles and roles of attention mechanisms, encoders, and word embeddings in the Transformer model. The fourth part will discuss and analyze the limitations of ChatGPT in machine translation. Finally, the fifth part will summarize the limitations of this research and the directions for future research.

2. Main Part

In 2017, Vaswani et al. first proposed the Transformer model [1], which efficiently processes sequential data based on self-attention mechanisms and positional encoding. The encoder-decoder structure of this model has achieved significant results in tasks such

as machine translation. Subsequent research by Radford et al. introduced the ChatGPT model based on the Transformer architecture, applying Transformer to dialogue generation tasks [2]. In 2020, Brown et al. introduced the concept of language models as few-shot learners, optimizing and exploring the pre-training mechanism of the ChatGPT model [1,2,3,15], providing new ideas for improving the model's performance and expanding its applications [3]. In the same year, Wolf et al. conducted research on self-attention mechanisms, proposing the Transformers model and achieving major progress in the field of natural language processing (NLP) [4]. This breakthrough provided new technical support for the improvement and application of the ChatGPT model. In this paper, we will use concepts such as dimensions, weighted vectors, CNN, and positional encoding. In the research context, dimensions usually refer to the size of the vector space in the model. In the Transformer model, the main dimensions include the word embedding dimension and the dimensions within the self-attention mechanism [5,7,10]. Before input words enter the Transformer model, each word needs to be mapped to a fixed-length word embedding vector. The word embedding dimension refers to the length of these word embedding vectors, which is typically set in the model's input layer. In the encoder and decoder of the Transformer, the self-attention mechanism is used to calculate the attention weights between each word and other words. During this process, the number of attention heads and the dimension of each attention head need to be defined, usually used for splitting and combining attention information. Weighted vectors typically refer to a set of weight values calculated by the attention mechanism, used to weight and aggregate different positions or words in the input sequence to produce context-aware feature representations. These weighted vectors frequently appear in the self-attention mechanisms of the encoder and decoder, capturing the importance of each position or word in the input sequence to effectively encode and decode sequence information. CNN [8,9] usually refers to convolutional neural networks, which are deep learning models commonly used for image and video processing. Compared to traditional neural networks, CNNs have better capabilities for processing images and sequential data because they can automatically learn features from images and extract the most useful information. In the Transformer model, in addition to word embeddings, positional embeddings are needed to represent the positions of words in a sentence. Because the Transformer does not adopt the RNN structure [12,13] and uses global information, it cannot utilize the order of words, which is crucial for NLP [2,3,11]. Therefore, the Transformer uses positional embeddings to save the relative or absolute positions of words in a sequence. Positional embeddings are denoted as PE, with the same dimensions as word embeddings. They can be calculated using the following formula:

$$PE_{(pos,2i)} = sin\ (pos/10000^{2i/d})$$

$$PE_{(pos,2i+1)} = \cos\ (pos/10000^{2i/d}) \quad [1]$$

In the Transformer model, taking "我有一本书" as an example, our goal is to translate it into English as "I have a book." However, the model cannot train directly on the Chinese "我有一本书." We need to convert it into numerical values before feeding it into the model. Generally, Chinese characters are converted into numerical values through word embeddings [6], either by training through a network or by using pre-trained models to transform them into continuous vectors. However, if too many Chinese characters are input, the resulting dimensions will be too large, consuming system memory. Each character is then formed into a continuous vector, assuming the embedding dimension is 5, using a 5-dimensional vector to represent each character. Positional encoding determines the position of words within a sentence. The attention mechanism in the Transformer model typically comprises self-attention mechanisms and multi-head attention mechanisms. The self-attention mechanism captures dependencies between different positions in the input sequence. Its basic idea is to calculate the correlation between each position and other positions in the input sequence, determining the importance of each position based on these correlations to achieve global attention to the sequence. The calculation process of the self-attention mechanism includes calculating the corresponding query (q), key (k), and value (v) vectors from the input vector x. The attention scores are calculated by taking the dot product of the query vector with the key vectors of other words. These attention scores are then normalized using Softmax, and the normalized scores are multiplied by the value vectors to obtain weighted representation vectors. After passing through the attention mechanism, several new encoded vectors are obtained. The encoder consists of multi-head attention mechanisms, Add & Norm, and Feed Forward layers [1]. The input to the first encoder block is the embedded vector of the Chinese "我有一本书," with dimensions [number of words, embedding dimension], i.e., [4,5]. After feature extraction through the multi-head attention mechanism, multiple output Z vectors are obtained, with each head extracting different patterns into matrices of dimensions [number of words, hidden vector dimension]. These Z matrices are then concatenated to form a large matrix of dimensions [number of words, number of heads * hidden vector dimension]. This matrix is fed into a fully connected network and mapped into a Z matrix with the same shape as the input vector to maintain consistency between the final output vector and the input dimensions, as the Transformer adopts a residual structure for ease of addition. The final Z matrix, after feature extraction through the multi-head attention mechanism, represents the encoded vector of each word. Add & Norm comprises two parts: a residual structure and LayerNorm. Feed Forward is a regular fully connected network with two layers.

To analyze the limitations of ChatGPT in machine translation, we invoked the API [14] to have GPT perform the task of mechanical translation and manually verified its translations to identify issues as shown in Figure 1 and Figure 2. After identifying the problematic areas, we analyzed which part of the Transformer model might be causing the issue, as illustrated below:

```python
import requests

api_url = 'https://api.132006.xyz/v1/chat/completions'
api_key = 'sk-slppzay6lcMrjgDF5285659835874f1491676475941lCe60'

headers = {
    'Content-Type': 'application/json',
    'Authorization': f'Bearer {api_key}'
}

request_data = {
    'messages': [
        {'role': 'system', 'content': '你是一个英汉互译助手.我发生英文您翻译为中文，发生中文翻译为英文，只需要输出结果。'},
        {'role': 'user', 'content': '你好'}
    ],
    'stream': False,
    'model': 'gpt-3.5-turbo',
    'temperature': 0.5,
    'presence_penalty': 0,
    'frequency_penalty': 0,
    'top_p': 1
}

response = requests.post(api_url, headers=headers, json=request_data)
data = response.json()

# 处理响应数据
print(data)
```

Figure 1. In this code, GPT acts as an English-Chinese translation assistant.

```
C:\Users\huawei\PycharmProjects\openai_test\.venv\Scripts\python.exe C:\Users\huawei\PycharmProjects\openai_test\openai_test1.py
{'choices': [{'content_filter_results': {'hate': {'filtered': False, 'severity': 'safe'}, 'self_harm': {'filtered': False, 'severity': 'safe'}, 'sexual': {'filtered': False, 'severity':
 'safe'}, 'violence': {'filtered': False, 'severity': 'safe'}}, 'finish_reason': 'stop', 'index': 0, 'logprobs': None, 'message': {'content': 'Hello!', 'role': 'assistant'}}], 'created':
 1714228493, 'id': 'chatcmpl-9IdQbe2ABuWl3iKyR1XgWKxZBaq5C', 'model': 'gpt-35-turbo', 'object': 'chat.completion', 'prompt_filter_results': [{'prompt_index': 0, 'content_filter_results':
 {'hate': {'filtered': False, 'severity': 'safe'}, 'self_harm': {'filtered': False, 'severity': 'safe'}, 'sexual': {'filtered': False, 'severity': 'safe'}, 'violence': {'filtered': False,
 'severity': 'safe'}}}], 'system_fingerprint': 'fp_2f57f81c11', 'usage': {'completion_tokens': 2, 'prompt_tokens': 60, 'total_tokens': 62}}

Process finished with exit code 0
```

Figure 2. When the user requests the translation of "你好" ("Hello"), GPT provides the correct answer "Hello!"

However, when the user inquires about polysemous words, GPT fails to provide the correct meanings, as shown below:

As shown in Figure 3,"run" in English has translations such as "跑步" (running), "运行" (operation), "进行" (proceed), and "一连串" (a series). GPT only provides the translation "跑步" (running).

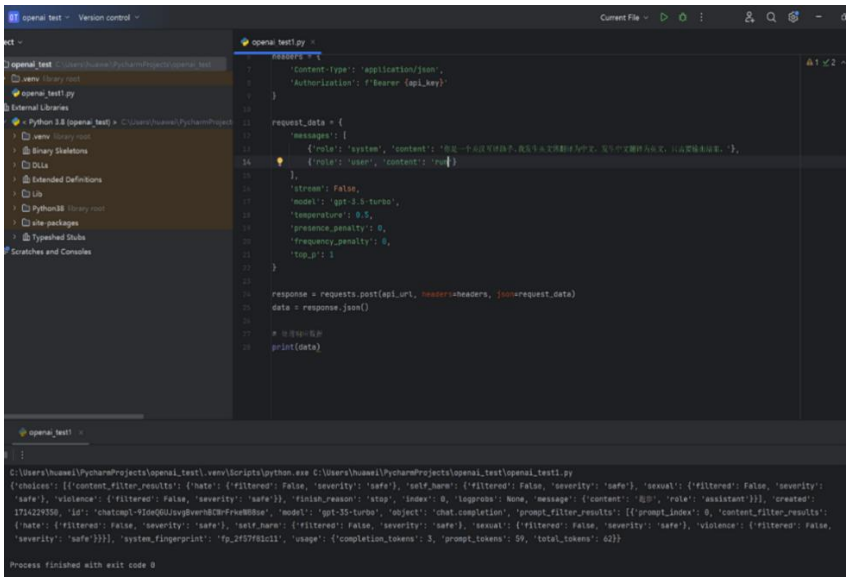

Figure 3. GPT received a request to translate "run" and responded accordingly

As shown in Figure 4,"light" in English has translations such as "光" (light), "灯光" (lamp), "轻的" (light in weight), and "亮的" (bright). GPT only provides the translation "光"(light).

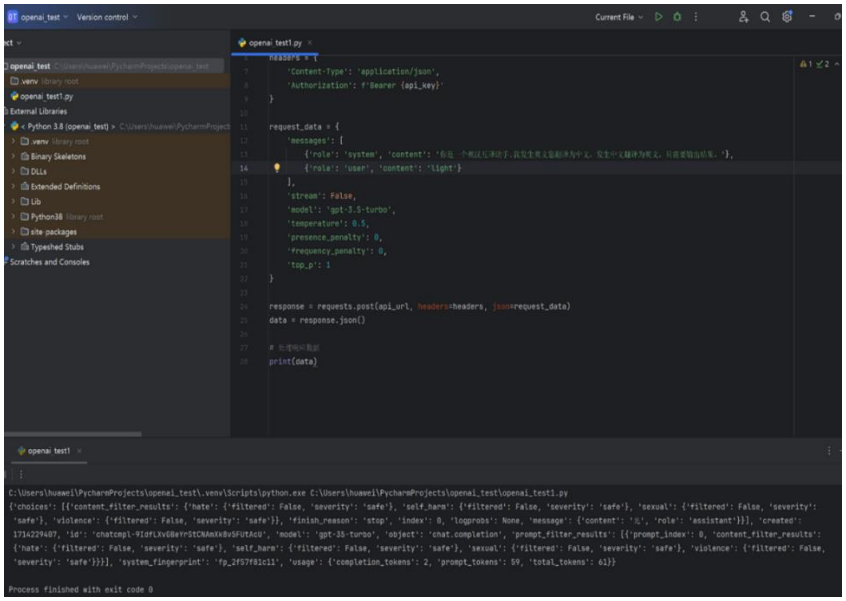

Figure 4. GPT received a request to translate "light" and responded accordingly

As shown in Figure 5,"bat" in English has translations such as "蝙蝠" (bat), "球棒" (bat), and "拍打" (to bat). GPT only provides the translation "蝙蝠" (bat).

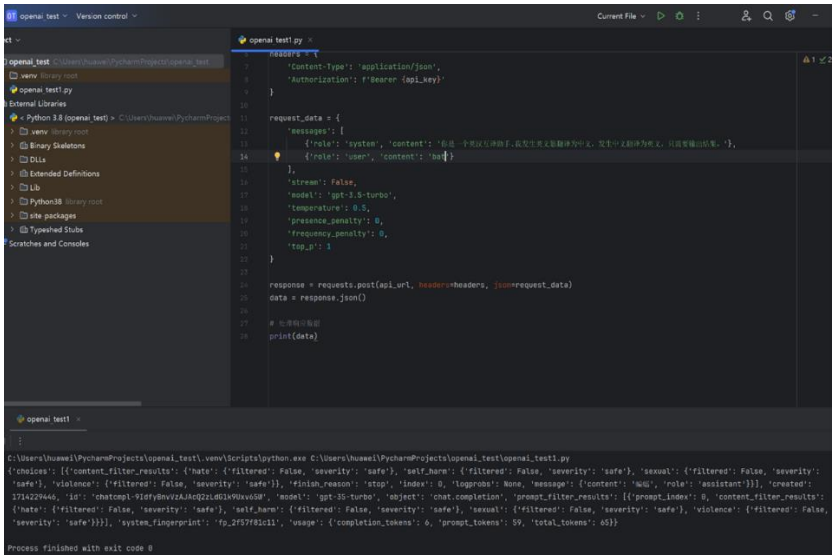

Figure 5. GPT received a request to translate "bat" and responded accordingly

Conversely, when the user inquiries about a Chinese word that can be translated into multiple English words, GPT fails to provide multiple English translations, as shown below:

As shown in Figure 6,"伟大的" has translations such as "great," "grand," "mighty," and "prodigious." GPT only provides the translation "great."

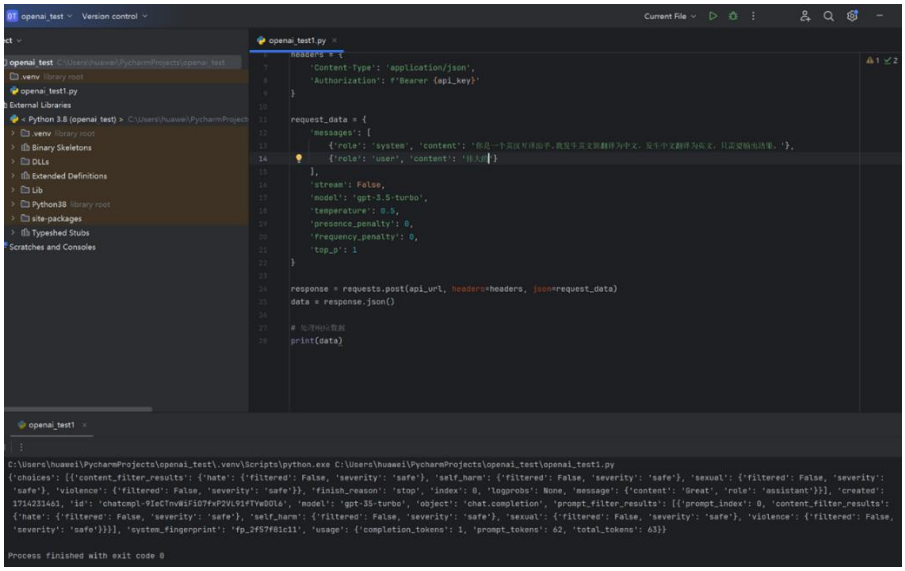

Figure 6. GPT received a request to translate "伟大的" and responded accordingly

As shown in Figure 7,"特别的" has translations such as "special," "particular," and "especial." GPT only provides the translation "special."

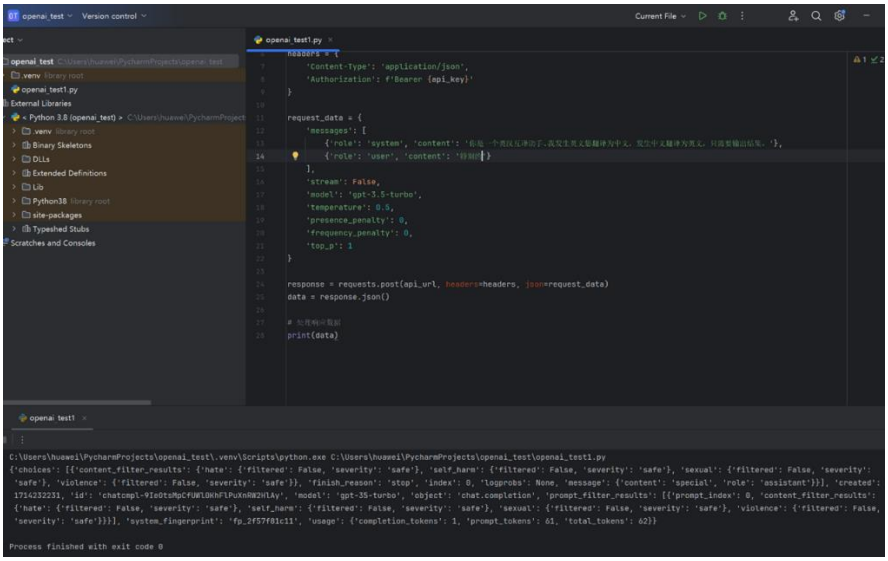

Figure 7. GPT received a request to translate "特别的" and responded accordingly

Furthermore, when the user inquiries about a technical term, GPT may not provide the correct answer, as shown below:

As shown in Figure 8,"CNN" (Convolutional Neural Network) is correctly translated as "卷积神经网络," but GPT does not provide the desired answer.

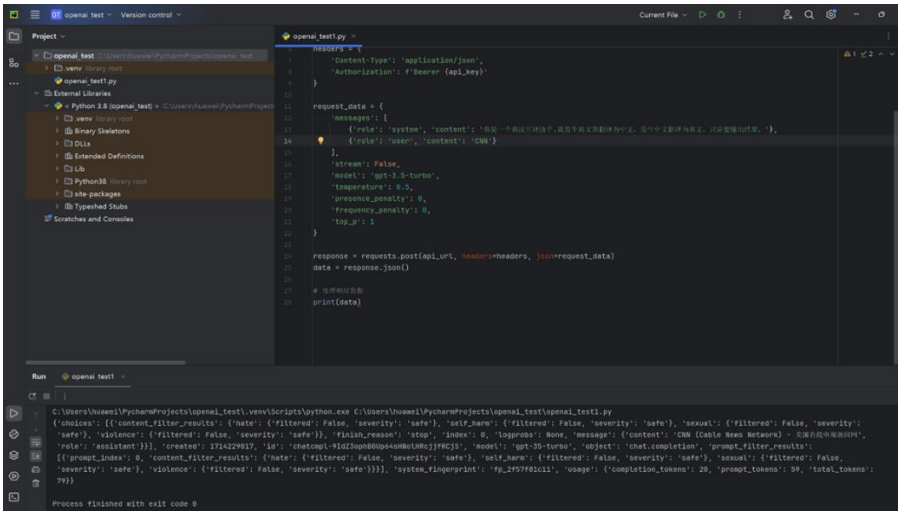

Figure 8. GPT received a request to translate "CNN" and responded accordingly

Based on the manual verification results and the initial Transformer model, it is speculated that the issues arise from the following parts:

1. Attention Mechanism: The Transformer model uses attention mechanisms to understand the context of the input text and weigh relevant parts during translation. If the correct meaning of a polysemous word is not adequately attended to in the context, the model may choose the wrong meaning for translation.

2. Encoder and Decoder: The encoder is responsible for encoding the input text into abstract representations, while the decoder generates the translation based on the encoder's output. If the encoder fails to accurately capture the semantic information of polysemous words in the input text, or if the decoder does not consider the correct meaning in the context while generating the translation, it can lead to inaccurate translations.

3. Word Embeddings: The Transformer model processes words by converting them into vector representations, which may contain ambiguities in representing word meanings, especially for polysemous words. If the word embedding vectors do not accurately capture the different meanings of polysemous words, the model may select the wrong meaning for translation.

4. Training Data: The training data for the Transformer model might be insufficient to cover all semantics and contexts, particularly for polysemous words common in specific fields or cultures. If the training data does not accurately or sufficiently annotate polysemous words, the model may fail to learn their correct meanings, resulting in inaccurate translations.

3. Conclusion

In summary, this paper provides an in-depth exploration of data intelligent generation and analysis based on ChatGPT. By reviewing and analyzing the concepts and components of the Transformer model, we have gained a deep understanding of the principles and roles of attention mechanisms, encoders, decoders, and word embeddings, thereby laying a theoretical foundation for subsequent research on ChatGPT. Additionally, this paper's main contribution is identifying the limitations of ChatGPT in machine translation tasks. Specifically, we analyzed issues such as ChatGPT's inadequate handling of polysemous words and limited understanding of domain-specific technical terms from the perspectives of attention mechanisms, encoders, decoders, and word embeddings based on the initial Transformer model. This analysis provides insights and directions for future research, such as improving the ChatGPT model to enhance its ability to handle polysemous words and better understand and generate language in specific domains. Additionally, exploring the combination of other models or methods, such as reinforcement learning and transfer learning, can further optimize ChatGPT's performance in data intelligent generation and analysis tasks. Finally, strengthening the interpretability and explainability of the model will make the generated results more credible and actionable.

The limitations of this research are mainly as follows: First, we focused primarily on the performance and limitations of ChatGPT in machine translation tasks, with relatively little exploration of other data intelligent generation and analysis tasks. This limits our comprehensive understanding of ChatGPT's applicability and effectiveness in other tasks. Second, our research concentrated only on the performance analysis of the current model version, without in-depth discussion on the evolution and improvement of the model. As technology advances and the model iterates, the performance and functionality of ChatGPT and its derivative models may improve, but this study did not delve into these aspects. Third, when analyzing the limitations of ChatGPT, we focused mainly on the structure and functionality of the model itself, paying less attention to external factors influencing model performance, such as data quality and training strategies. Therefore, future research can expand the scope by considering more tasks and application scenarios, focusing on the evolution and improvement of the model, and updating research findings promptly.

Additionally, incorporating external factors affecting model performance can comprehensively evaluate the practical effectiveness and application prospects of data intelligent generation and analysis based on ChatGPT.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., & Gomez, A. N., et al. (2017). Attention is all you need. *arXiv*. https://arxiv.org/abs/1706.03762

[2] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI*. https://www.openai.com/research/language-understanding-generative-pretraining

[3] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., & Amodei, D. (2020). Language models are few-shot learners. *arXiv*. https://arxiv.org/abs/2005.14165

[4] Sun, B., & Li, K. (2021). Neural dialogue generation methods in open domain: A survey. *Natural Language Processing Research, 1*(3-4), 56-70. https://doi.org/10.2991/nlpr.d.210715.001

[5] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020, October). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (pp. 38-45). https://doi.org/10.18653/v1/2020.emnlp-demos.6

[6] Kalyan, K. S., Rajasekharan, A., & Sangeetha, S. (2021). Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*. https://arxiv.org/abs/2108.05542

[7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. https://arxiv.org/abs/1810.04805

[8] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60*(6), 84-90. https://doi.org/10.1145/3065386

[9] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data, 8*, 1-74. https://doi.org/10.1186/s40537-021-00444-8

[10] Shi, L., Wang, Y., Cheng, Y., & Wei, R. B. (2020). A review of attention mechanism in natural language processing. *Data Analysis and Knowledge Discovery, 4*(5), 1-14. https://doi.org/10.11925/infotech.2096-3467.2019.0462

[11] Xu, G., & Wang, H. F. (2011). Development of topic models in natural language processing. *Journal of Computer Science and Technology, 34*(8), 1423-1436. https://doi.org/10.1007/s11390-011-1193-3

[12] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. https://arxiv.org/abs/1406.1078

[13] Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. *arXiv preprint arXiv:1702.01923*. https://arxiv.org/abs/1702.01923

[14] Gu, X., Zhang, H., Zhang, D., & Kim, S. (2016, November). Deep API learning. In *Proceedings of the 2016 24th ACM SIGSOFT international symposium on foundations of software engineering* (pp. 631-642). https://doi.org/10.1145/2950290.2950334

[15] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. https://arxiv.org/abs/1810.04805