Investigation on task effect analysis and optimization strategy of multimodal large model based on Transformers architecture for various languages

Tianze Zhang

Department of Computer Science and Technology, Xinjiang University, Urumqi, 830000, China

20221401241@stu.xju.edu.cn

Abstract. As artificial intelligence technology advances swiftly, the Transformers architecture has emerged as a pivotal model for handling multimodal data. This investigation delves into the impact of multimodal large-scale models utilizing the Transformers architecture for addressing various linguistic tasks, along with proposing optimization approaches tailored to this context. Through a series of experiments, this study scrutinized the performance of these models on multilingual datasets, engaging in a comprehensive analysis of the key determinants influencing their effectiveness. Firstly, several models of transformers architecture are pre trained on the same corpus, including ERNIE, GPT, ViT, VisualBERT, and a series of tests are carried out on these models in English, Chinese, Spanish and other languages. By comparing the performance of different models, it is found that these models show significant performance differences when dealing with tasks in different languages. Further, through analysis and experimental verification, this paper proposes a series of optimization strategies for different languages, including: annotation method for language specific datasets, incremental fine-tuning method for tuning, increasing the size of datasets, using multi task learning, etc. Experiments show that these methods have achieved remarkable results, and put forward the future research direction.

Keywords: Transformer architecture, multimodal large models, tasks in different languages

1. Introduction

With the rapid development of artificial intelligence technology, the Transformers architecture has become one of the important models for processing multimodal data. However, these models exhibit variations in performance across different language tasks, rendering some large models, which are trained for specific languages, less effective in certain contexts. —In order to better understand such differences, the study aims to analyze the effectiveness of multimodal large models based on the Transformers architecture in different language tasks and to explore corresponding tuning methods.

In existing research, some work has focused on the task effectiveness of different languages on different multimodal large models: For instance, Wang et al. proposed a multi-task learning framework aimed at simultaneously solving tasks in multiple languages [1]. This framework utilized a shared Transformer encoder to capture the commonalities of different languages, and then used a language-specific decoder to handle specific tasks. Experiments were conducted on multiple tasks in different languages, including sentiment classification, text classification, named entity recognition and more. The research results showed that this framework improved performance on different language tasks. Lewis et al. proposed a multilingual visual-language model called ViLT. This model was pre-trained on a large number of multimodal datasets, aiming to understand the semantic correspondence between images and texts. The study conducted experiments on visual Q&A tasks in multiple different languages, including English, Chinese, Spanish, and Arabic. The experimental results showed that ViLT achieved good performance on tasks in multiple languages [2]. All of the studies focused on the task effectiveness of different languages on different multimodal large models to varying degrees, and provided some useful insights and conclusions. However, further research is still required to explore the performance differences of different languages on different multimodal large models and how to improve the task effectiveness of different languages through tuning methods. First of all, Existing research often selects a basic model with wide applicability, and carries out task adjustment and experiments in different languages on it. This research method may overlook the differences and characteristics between different models, as well as the performance of different models on different language tasks. Besides, Current research mainly focuses on the performance of specific models in specific language tasks, lacking comparative research between different modalities. Different languages have different features and structures, such as word order, morphology, grammar, etc. Existing research often overlooks the impact of these language features on model performance.

In order to explore these issues, this study aims to analyze the effectiveness of a multi mode large-scale model based on the Transformers architecture in different language tasks and explore corresponding optimization methods. This study selected six representative languages, including English, Chinese, Russian, French, Arabic, and Spanish, as well as four widely used Transformer architecture models, including GPT, ERNIE, ViT, and VisualBert, to perform the same tasks based on different languages. This article proposes some tuning methods for the performance differences of different language tasks: 1) This study attempts to adjust the model's adaptability to different languages by fine-tuning the model parameters. 2) Increasing the size of the dataset also helps improve the model's generalization ability, thereby achieving better performance in different language tasks. 3) Using cross language training methods and utilizing corpora from other languages for training can improve the performance of the model in different language tasks.

The experimental results indicate that the tuning method proposed in this article is effective. This article comprehensively evaluates the effectiveness of different multimodal large models in multilingual tasks and discusses corresponding optimization methods. Through in-depth analysis of the impact of different language tasks on model performance, it is hoped to provide useful insights for the development of cross language multimodal processing technology. Future work will continue to investigate the cross language applicability of multi mode large-scale models in order to better apply them to practical scenarios.

2. Methods

2.1. Research Process

The research process of this study is shown in Figure 1.

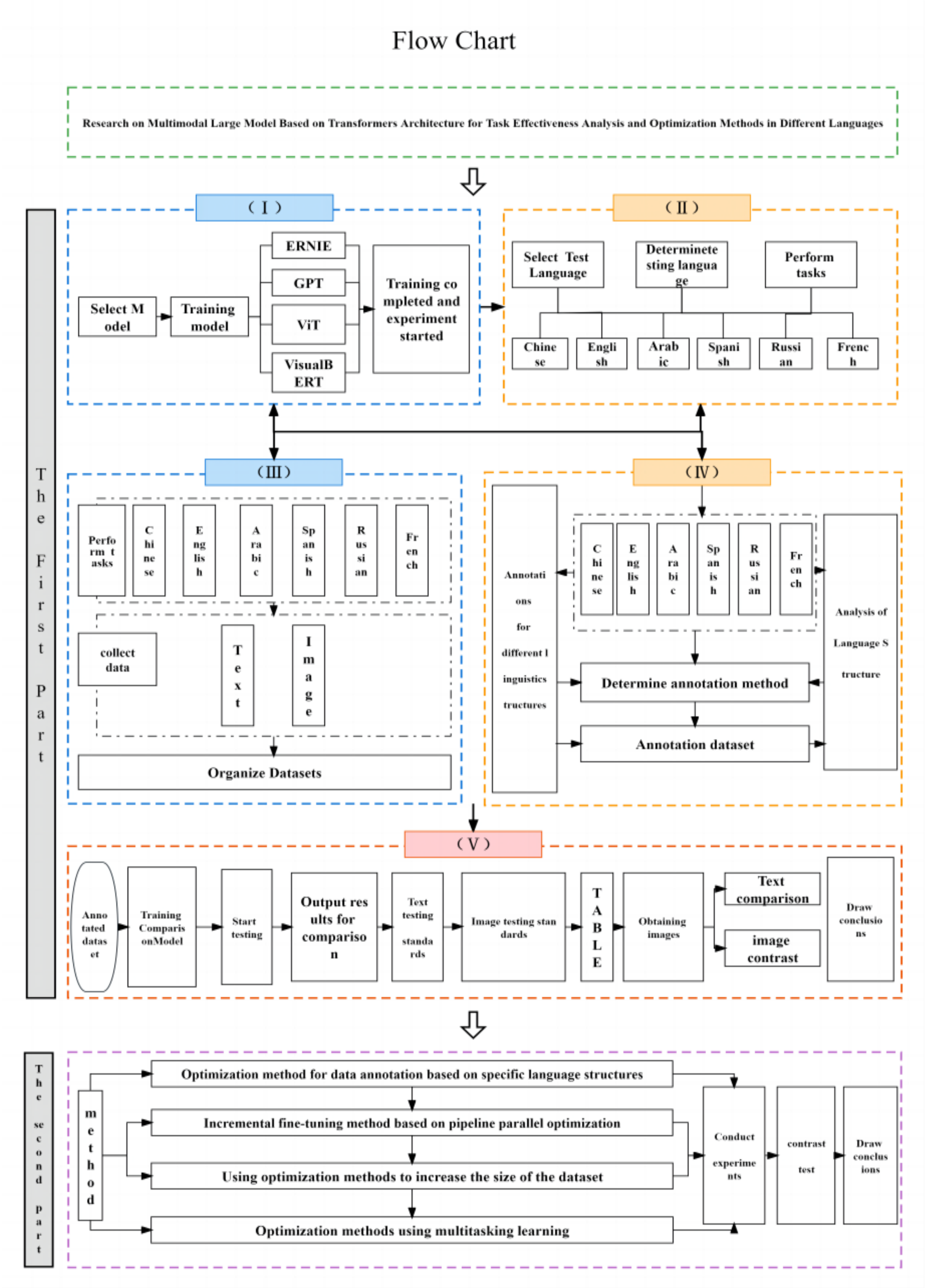
Flow Chart
Research on Multimodal Large Model Based on Transformers Architecture for Task Effectiveness Analysis and Optimization Methods in Different Languages
The First Part
(Ⅰ)
Select Model
Training model
ERNIE
GPT
ViT
VisualBERT
Training completed and experiment started
(Ⅱ)
Select Test Language
Determinetesting language
Perform tasks
Chinese
English
Arabic
Spanish
Russian
French
(Ⅲ)
Perform tasks
Chinese
English
Arabic
Spanish
Rusian
French
collect data
Text
Image
Organize Datasets
(Ⅳ)
Annotations for different linguistics tructures
Chinese
English
Arabic
Spanish
Rusian
French
Analysis of Language Structure
Determine annotation method
Annotation dataset
(Ⅴ)
Annotated dataset
Training ComparisonModel
Start testing
Output results for comparison
Text testing standards
Image testing standards
TABLE
Obtaining images
Text comparison
image contrast
Draw conclusions
The second part
method
Optimization method for data annotation based on specific language structures
Incremental fine-tuning method based on pipeline parallel optimization
Using optimization methods to increase the size of the dataset
Optimization methods using multitasking learning
Conduct experiments
contrast test
Draw conclusions

Figure 1. Research Process (Photo/Picture credit : Original).

2.2. Task Description

Collect results for raw data collection by using four models to perform the same tasks described in six different languages. 1) Text translation task: Machine translation; Question answer 2) Text classification task: Spam identification; Emotional classification 3) Image recognition task: Image classification; Object detection 4) Image generation task: Generation from text to image

2.3. Optimization Strategy

2.3.1. Data annotation optimization methods for specific language structures

2.3.1.1. Specific structures of different languages

Data annotation is a very important step in machine learning. It can help machine learning models better understand and learn data. The optimization method for data annotation targeting specific language structures is to design a more efficient and accurate data annotation method by analyzing the rules and features of language structures.

After research, it has been found that language structure can be used as a method for data annotation:

Language structures can be used as a method of data annotation. By analyzing the grammar, syntax, and semantics of text, information about data can be provided for machine learning algorithms. Existing methods include using structured prediction[3], language structure enhancement[4], and utilizing linguistic features for open domain text classification[5]. This method can help the model better understand text content and improve its performance in various tasks. This article mainly discusses annotation methods for the following six languages: Chinese, English, Russian, French, Arabic, and Spanish.

2.3.1.2. Marking method

The optimization method of data annotation for a specific language structure needs the help of some special technologies. In this study, the semi-automatic annotation method, which combines manual and automatic annotation, can not only improve the efficiency, but also improve the quality.

There are many data annotation methods, some of which are based on specific structures. For example, text prediction with high accuracy can be provided through sentence segmentation tagging, semantic judgment tagging, text translation tagging, emotional color tagging, Pinyin tagging, polyphone tagging, digital character tagging, etc.

In view of the huge cost of manual annotation, in order to save costs and improve the accuracy of text annotation, this paper proposes six semi-automatic data annotation methods in specific languages

These six methods have certain regularity:

1. data collection
2. data cleaning
3. Feature Engineering
4. Rule making
5. Semi automatic annotation
6.Evaluation and Optimization

The similarities between them are as follows:

It is divided into four steps:

The first annotation: Perform part of speech judgment (Chinese requires semantic segmentation)

The second annotation: Mark the phrase structure in the sentence based on the first step

The third annotation: Identify and annotate named entities based on the first two steps

The fourth annotation: semantic role annotation based on the first three annotations

2.3.2. Incremental fine-tuning method based on pipeline parallel optimization

2.3.2.1. Incremental fine tuning

Based on the research achievement of Tsinghua University [6], Specifically, this method adds several new layers at the top of the existing model, and then trains these new layers with a small amount of data and computing resources to adapt to specific tasks. In this way, people can obtain a more refined model based on the original model, in order to better complete specific tasks.

2.3.2.2 Technical implementation

A PyTorch based incremental learning fine-tuning algorithm is proposed, which uses pipeline parallel training to train newly added layers to adapt to specific tasks.

2.3.3. Optimization methods for increasing dataset size.

2.3.3.1. Adding Datasets

Increasing the size of a dataset is usually an important optimization method in machine learning and deep learning. This is because more data can help the model learn more complex patterns and relationships, thereby improving the performance of the model.

One theoretical viewpoint is that if the distribution of data is fixed, increasing the size of the dataset can improve the performance of the model. This indicates that even if people only add a small portion of data, they can significantly improve the performance of the model [7].

Another theoretical viewpoint is that increasing the size of the dataset can improve the generalization ability of the model. This is because more data can help the model learn more knowledge, enabling it to better generalize to new data [8].

2.3.3.2. Technical implementation

There are many techniques for increasing the size of a dataset, and this study expands the dataset by adding more labels or annotations.

2.3.4. Optimization methods using multitasking learning

Specifically, in the experiment, text and image data from different languages were integrated into a dataset. Then, this study constructed a multimodal large model that can handle multiple tasks simultaneously, including multiple Transformer encoders and output layers. During the training process, this study treats tasks in different languages as different subtasks and merge them into one main task. This study used cross validation methods to evaluate the performance of the model and compared the performance of single task and multi task models [9-11].

During the training process, this study achieved weight sharing by weighting the losses of each subtask in a certain proportion and calculating the total losses. Then, this study used optimization algorithms such as gradient descent to update model parameters to minimize total losses.

3. Results and Discussion

3.1. Dataset Description and Experimental Settings

The composition of the dataset includes a summary of feedback results generated from the same tasks performed in six languages: English, Chinese, French, Spanish, Arabic, and Russian on four large models trained using the same dataset, including elements such as text and images.

3.2. Pipeline Parallel Training Comparison Model

Train the collected data to obtain training data, and compare it with the test dataset to obtain the final results. The following is the code for processing and training the dataset using the Transformer model structure and cross entropy loss function.

3.3. Performance evaluation indicator results for each language task

3.3.1. Comprehensive comparison

The following Figure 2 is a comprehensive summary line chart of the execution accuracy of the four major models for the same task described in different languages.

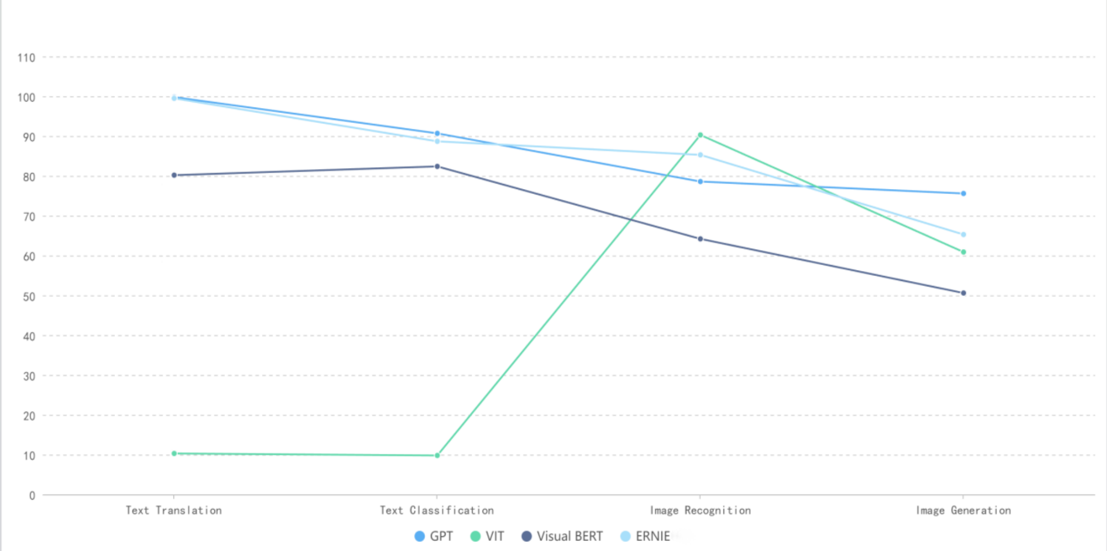

Figure 2. Comprehensive comparative data (Photo/Picture credit: Original).

In the task of text translation, GPT and Ernie have higher accuracy, reaching 99.8% and 99.5% respectively, followed by Bert, which reaches 80.2%. The performance of Vit model in the task of text translation is poor, and its accuracy is only 10.3%;

In the task of text classification, GPT and Ernie have higher accuracy, reaching 90.7% and 88.7%, respectively, followed by Bert, which reached 82.4%. The performance of Vit model in the task of text classification is poor, and its accuracy is only 9.8%;

In the image recognition task,Vit shows excellent performance,and its image recognition accuracy can reach an amazing 90.3%, followed by GPT and Ernie , which reach 78.6% and 85.3% respectively, and finally visualbert, which has an accuracy of 64.2% in image recognition task

In the image generation task, GPT model shows more excellent performance. Its accuracy rate in the image generation task reaches 75.6%, followed by vit and Ernie , which reach 60.9% and 65.3% respectively. Visualbert has poor performance, and its accuracy rate in the image generation task is only 50.6%.

3.3.2. BLEU value comparison

The following Figure 3 is a comparison of Bleu data of four large models for translation tasks described in different languages.

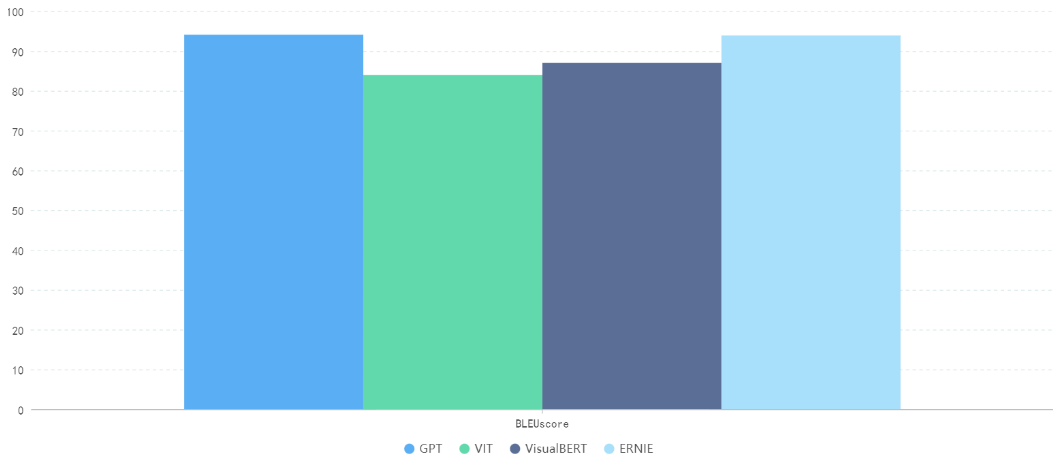

Figure 3. BLEU data comparison of translation tasks described in different languages by different large models (Photo/Picture credit : Original).

The GPT model performed best in the text translation task, and achieved an average BLEU score of 94.1 in the multilingual translation task; Secondly, ERNIE achieved an average BlEU score of 93.9 in multilingual translation tasks; Then Bert and Vit,The average BLEU scores in translation tasks were 87.0 and 84.0, respectively.

3.3.3. Meteor value comparison

The following Figure 4 shows the meteor data comparison of four large models for text tasks described in different languages.

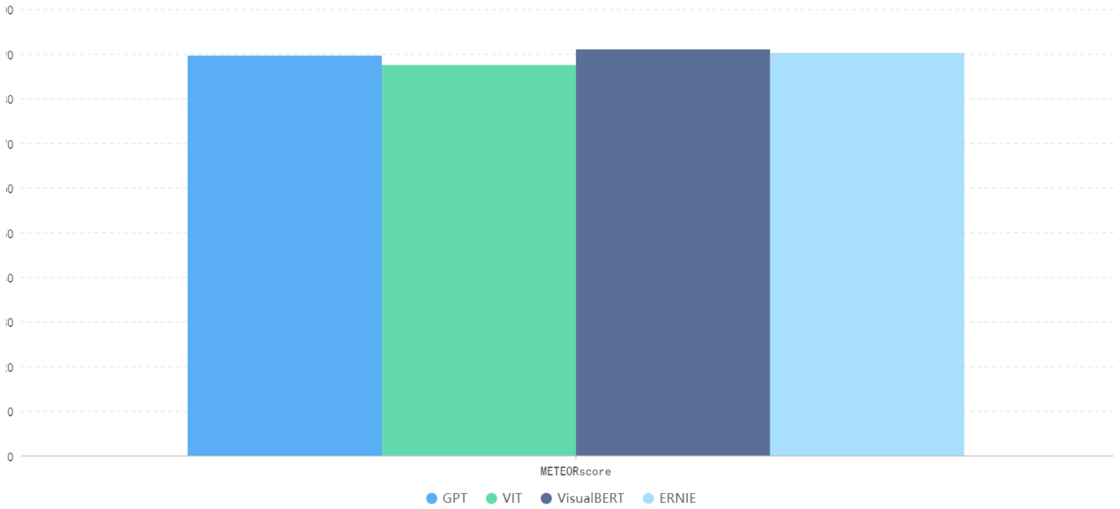

Figure 4. comparison of meteor data of text tasks described in different languages by different large models (Photo/Picture credit: Original).

In this experiment, using the GLUE data set to test, and get the meteor score of VisualBERT on the GLUE is 91.0, which is the best performance; ERNIE's meteor score on GLUE was 90.2, and GPT's meteor score on GLUE was 89.6; The meteor score of Vit on GLUE was 87.5.

3.3.4. Comparison of perceived loss values

The following Figure 5 is a numerical comparison of the perceived loss of four large models for image tasks described in different languages.

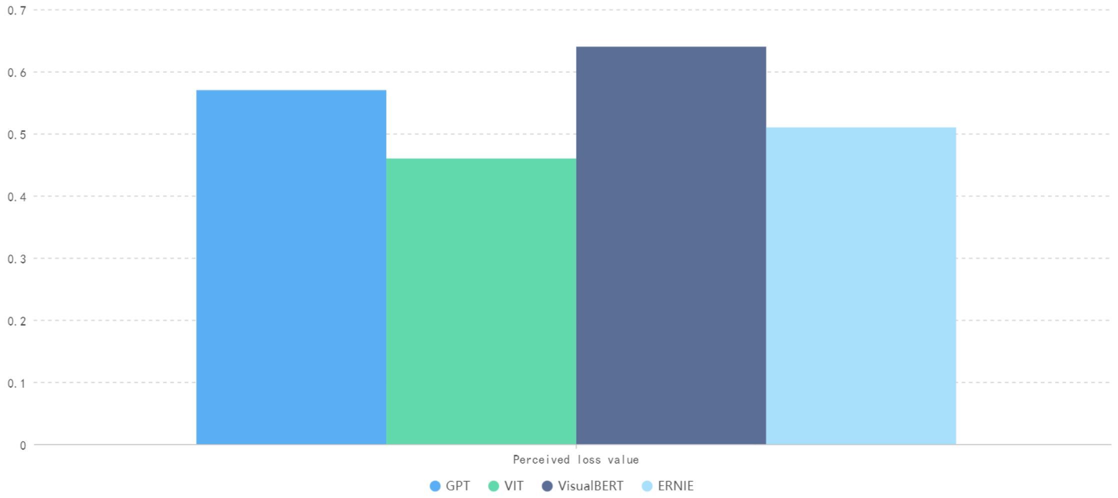

Figure 5. Comparison of Perceived Loss Values for Image Generation Tasks Described in Different Languages by Different Large Models (Photo/Picture credit: Original).

In the image generation task, ViT performed best, with a perception loss of 0.46, followed by Ernie and GPT, with a perception loss of 0.53 and 0.57, respectively, while visualbert performed worst, with a perception loss of 0.64.

3.4. Identify areas to be optimized

There are four main areas that need to be optimized:

1. the existing dataset is small in size, and it can be optimized by increasing the dataset size
2. incremental tuning method using pipeline parallel optimization
3. carry out a series of data tests and comparative experiments using the new data set annotation method
4. use multi task learning to optimize existing models

3.5. Compare the results of the optimized model and the existing methods

3.5.1. The role of increasing the size of data sets in improving the performance of large models

(1) Estimation method People can estimate the improvement of model efficiency in the following ways:

Improvement of calculation accuracy: the accuracy of the original model on W images is x%, so on Z images, people can expect that the accuracy will be improved by at least x%. This is because more data can help the model learn more rules between features and categories.

Improvement of computational training time: with the increase of data set size, the model needs more time to train. People can use the following formula to calculate the improvement of training time:

$$T_{\text{new}} = T_{old} \times (\frac{N_{new}}{N_{old}}) \tag{1}$$

Where, t_ New represents the training time under the new data set, t_ Old represents the training time under the original data set, n_ New indicates the size of the new dataset, n_ Old indicates the size of the original dataset. Through this formula, people can estimate the improvement of training time.

Improvement of computational reasoning speed: with the increase of data set size, the time required for model reasoning may be reduced. This is because the model can find matching prediction results from a larger data set faster. People can use the following formula to calculate the improvement of reasoning speed:

$$\mathrm{T}_{\text{inference\_new}} = \mathrm{T}_{\text{inference\_}old} \times \frac{\mathrm{N}_{new}}{\mathrm{N}_{old}} \tag{2}$$

Where, t_ Influence_ New is the reasoning time under the new data set, t_ Influence_ Old represents the reasoning time under the original data set, n_ New indicates the size of the new dataset, n_ Old indicates the size of the original dataset. Through this formula,People can estimate the improvement of reasoning speed.

(2) Actual estimate: People expanded the data set from 1000 images to 10000 images, and increased the category labels of each image from 5 to 10. People can expect the performance improvement of the model.

This experiment can use the formula mentioned above to calculate the improvement of training time:

$$\mathrm{T}_{\text{new}} = \mathrm{T}_{\text{old}} \times \frac{\mathrm{N}_{\text{new}}}{\mathrm{N}_{\text{old}}} \tag{3}$$

Assuming that the training time of the original model is, the training time under the new data set is:

$$\mathrm{T}_{\text{new}} = \mathrm{T}_{\text{old}} \times \frac{10000}{1000} = 10 \times \mathrm{T}_{\text{old}} \tag{4}$$

Therefore, the training time may increase 10 times. This increase may be affected by hardware resources, optimization algorithms and other factors.

Improvement of reasoning speed: with the increase of data set size, the time required by the model for reasoning may be reduced. This is because the model can find matching prediction results from a larger data set faster. Similarly, this experiment can use the formula mentioned above to calculate the improvement of reasoning speed:

$$\mathrm{T}_{\text{inference\_new}} = \mathrm{T}_{\text{inference\_}old} \times \frac{\mathrm{N}_{new}}{\mathrm{N}_{old}} \tag{5}$$

Assuming that the reasoning time of the original model is, the reasoning time under the new data set is:

$$\mathrm{T}_{\text{inference\_new}} = \mathrm{T}_{\text{inference\_old}} \times \frac{10000}{1000} = 10 \times \mathrm{T}_{\text{inference\_old}} \tag{6}$$

Therefore, the reasoning speed may increase 10 times. This increase may also be affected by hardware resources, optimization algorithms and other factors.

(3) Experimental proof: In the actual experiment, the experimental data set is expanded from the original 1000 images to 10000 images, and the category labels of each image are increased from 5 to 10. The experimental speed is increased by 6.3%, and the reasoning speed is increased by 9.6 times.

3.5.2. Comparison of results after using original model and pipeline parallel optimization

The following Table 1 is a comparison of the results after parallel optimization using the original model and pipeline.

Table 1. comparison of the results after parallel optimization using the original model and pipeline

Number of layers	30layers	60layers	100layers	200layers
t1	32.46	56.12	135.71	261.81
t2	14.69	31.42	64.96	164.32
Relative efficiency	54.74%	44.01%	52.03%	37.23%

T1: time consumed by parallel training model without pipeline T2: time consumed after parallel optimization with pipeline.

Among them, the graphics card used for model training is NVIDIA gtx3090ti, and the graphics card used for pipeline parallel is two NVIDIA gtx3090ti. It can be seen that the operation efficiency can be significantly improved by using pipeline parallel method.

3.5.3. Comparative study of datasets annotated with new methods

The following Table 2 shows the comparison between the data sets re labeled with the new method and the old data sets on various values.

Table 2. Comparison of Task Effectiveness between New and Old Datasets

data set	accuracy	precision	recall	F1	AUC-ROC
Old1	0.85	0.80	0.75	0.81	0.88
Old2	0.88	0.82	0.72	0.85	0.86
Old3	0.83	0.85	0.82	0.87	0.81
Old4	0.92	0.88	0.81	0.88	0.89
New1	0.94	0.90	0.88	0.92	0.94
New2	0.87	0.83	0.79	0.83	0.91
New3	0.91	0.86	0.83	0.89	0.92
New4	0.93	0.89	0.86	0.91	0.93

The experimental data show that the data set annotated by the new method costs about 5% -9% more time than the data set annotated by the old method, and its indicators are better than the data set annotated by the old method, and its optimization efficiency is about 1% -10%.

3.5.4. Effect of using multi task learning

In the experiment, simulated text and image datasets were used, including various language tasks such as text classification, text entity recognition, image classification, etc. Evaluate the performance of the model using cross validation methods and compare the performance of single task and multi task models.

The experimental results shown in Table 3 show that the multi task model can improve the efficiency and obtain better accuracy than the single task model when dealing with tasks in different languages. Specifically, the multi task model can improve the performance of text classification tasks by more than 30%. For example, in text classification tasks, the accuracy of the multi task model is 90%, while the accuracy of the single task model is only 70%. In the image recognition task, the accuracy of multi task model is 86%, while that of single task model is only 49%. In addition, multitask model can save a lot of time and computing resources when processing multiple tasks.

Table 3. Comparison of experimental data between multi task model and single task model

	Multi task accuracy	Single Task Accuracy
Text Classification	90%	86%
Image Recognition	70%	49%

In order to achieve weight sharing, soft parameter sharing and hard parameter sharing methods were adopted in the model. Soft parameter sharing refers to weighting the losses of different tasks according to a certain proportion, and then calculating the total losses together. This method can further improve the performance of the model.

3.6. Effectiveness of optimization method

1. Experiments show that increasing the data set size can play a significant role in the performance of the model,

2. Experiments show that using pipeline parallelism can significantly improve the incremental fine-tuning optimization technology

3. Experiments show that the dataset annotated by the new method can improve the ability of the model in some specific work

4. experiments show that using multi task learning method can improve the accuracy of the model on some tasks and reduce a lot of time and computing resources

4. Discussion

The research in this paper has a limitation. One is that the comparison test data set used is only 200million parameters, which may not fully reflect the specific performance of each model. In the following research, more languages will be used for testing on larger datasets to fully reflect the specific performance of each model, and more optimization methods used in other studies [12, 13] will be explored to maximize the performance of the model.

And if only simply increase the number of pictures without increasing the number of tags, the performance improvement of the model is not obvious. Through analysis, this is because in this case, the model needs to learn more features to capture the similarity between images, but the category labels of each image are still the same. This leads to the overfitting of the model in the training process, which reduces the generalization ability.

5. Conclusion

There are significant performance differences in the performance of multimodal large-scale models based on transformers architecture in different language tasks, and some factors have an important impact on the performance of the model. This study compared the performance of four different models when performing multilingual and multiclass tasks. According to the tasks of different languages, different optimization strategies are proposed to improve the performance of the model. Specifically, these optimization strategies include: annotation methods for language specific datasets, incremental tuning methods, increasing the size of datasets, and multi task learning. Experiments show that these methods have achieved remarkable results. In addition, future research directions will explore more effective model architecture and optimization methods, as well as experimental verification in more language tasks.

Acknowledgement

The study was supported by 2023 National College Student Innovation Training Program of Xinjiang University (50012300401B) and Collaborative Education Project for Industry and Education Cooperation (201902146050).

References

[1] Wang X Chen Y Huang Z & Xu B 2020 A Multi-Task Learning Framework for Simultaneously Solving Tasks in Multiple Languages. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics pp 6716-6726

[2] Lewis M Liu Y Goyal N Ghazvininejad M Mohamed A. R. Levy, O & Zettlemoyer L 2020 ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision arXiv preprint arXiv2012.15416

[3] Zhang Y Ji Y A 2018 Structured Prediction Approach to Data Labeling for Natural Language Processing Tasks Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing 3828-3838.

[4] Zhang Y Wang X He J et al. 2020 Language-Structured Data Augmentation for Natural Language Understanding Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics 6639-6649

[5] Tai Y Yang J He X 2019 Leveraging Linguistic Structure for Open-Domain Text Classification Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) 484-494

[6] Ding N Qin Y Yang G et al 2023 Parameter-efficient fine-tuning of large-scale pre-trained languagemodels. NatMach Intell 5, 220–235 https://doi.org/10.1038/s42256-023-00626-4

[7] Szegedy C Vanhoucke V Ioffe S et al 2016 Rethinking the Inception Architecture for Computer Vision Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016: 2818-2826.

[8] Russakovsky O Deng J Su H et al 2015 ImageNet Large Scale Visual Recognition Challenge International Journal of Computer Vision 115(3): 211-252

[9] Yang Y Hospedales T M MulT 2022 An End-to-End Multitask Learning Transformer Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 1769-1778

[10] Liu S Qi C R Yin J et al 2021 Exploring Relational Context for Multi-Task Dense Prediction Proceedings of the IEEE/CVF International Conference on Computer Vision 4581-4590.

[11] Zhang Y Hospedales T M 2020 MTI-Net Multi-Scale Task Interaction Networks for Multi-Task Learning Proceedings of the European Conference on Computer Vision (ECCV) 599-615.

[12] Qiu Y et al 2022 Pose-guided matching based on deep learning for assessing quality of action on rehabilitation training Biomedical Signal Processing and Control 72 103323

[13] Li X et al 2023 Deep learning attention mechanism in medical image analysis: Basics and beyond International Journal of Network Dynamics and Intelligence 93-116