System Report for CCL24-Eval Task 6: Essay Rhetoric Recognition and Understanding Using Synthetic Data and Model Ensemble Enhanced Large Language Models

Jinwang Song, Hongying Zan*, Kunli Zhang
School of Computer and Artificial Intelligence, Zhengzhou University
Zhengzhou, 450001
jwsong@gs.zzu.edu.cn, iehyzan@zzu.edu.cn, ieklzhang@zzu.edu.cn

Abstract

Natural language processing technology has been widely applied in the field of education. Essay writing serves as a crucial method for evaluating students' language skills and logical thinking abilities. Rhetoric, an essential component of essay, is also a key reference for assessing writing quality. In the era of large language models (LLMs), applying LLMs to the tasks of automatic classification and extraction of rhetorical devices is of significant importance. In this paper, we fine-tune LLMs with specific instructions to adapt them for the tasks of recognizing and extracting rhetorical devices in essays. To further enhance the performance of LLMs, we experimented with multi-task fine-tuning and expanded the training dataset through synthetic data. Additionally, we explored a model ensemble approach based on label re-inference. Our method achieved a score of 66.29 in Task 6 of the CCL 2024 Eval, Chinese Essay Rhetoric Recognition and Understanding(CERRU), securing the first position.

Keywords: Rhetoric Recognition, Large Language Models, Model Ensemble, Synthetic Data

1 Introduction

Essay writing is a crucial means of assessing students' language proficiency and logical thinking skills. The development and application of natural language processing (NLP) technologies have significantly advanced the field of education. Typically, grading essays demands substantial time and effort from teachers. By applying automation technologies to this process, we can alleviate the workload on teachers, allowing them to focus more on instruction and student guidance. Rhetorical devices are an essential component of essays, making it necessary to automate the extraction and classification of these devices as a key dimension in the automated assessment of writing quality.

Previous work in the fields of machine learning and deep learning has extensively explored the identification and classification of rhetorical devices in writing. For instance, (Xiaoxi et al., 2018) used convolutional neural networks and support vector machines to identify metaphors in both Chinese and English datasets. (Hu et al., 2017) employed sequential models to recognize metaphors in text, while (Liu et al., 2018) adopted a multi-task learning approach to classify rhetorical sentences and extract their rhetorical components. Additionally, (Li and Li, 2022; Iqbal et al., 2023) utilized BERT models to identify metaphors in compositions by overseas Chinese students.

The recognition and extraction of rhetorical devices can be fundamentally categorized as tasks of text classification and entity recognition, both of which have been extensively studied. The introduction of the GPT series (Radford et al., ; Radford et al., 2019; Brown et al., 2020) has sparked significant interest in LLMs within the NLP community, marking the advent of the era of large language models (Zhao et al., 2023). Following the release of the LLaMA series (Touvron et al., 2023), the open-source community for LLMs has flourished. In this new era, some studies have utilized the in-context learning capabilities of large models for text classification (Sun et al., 2023). Furthermore, (Wang et al., 2023) demonstrated the effectiveness of supervised instruction fine-tuning of large language models for information extraction tasks.

*Corresponding author

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 223-231, Taiyuan, China, July 25 - 28, 2024.
Volume3: Evaluations

Against this backdrop, we explored the use of generative large language models for rhetorical recognition and extraction in the CCL24-Eval Task 6 Chinese Essay Rhetoric Recognition and Understanding(CERRU). For this task, we processed the dataset into fine-tuning instructions and applied parameter-efficient fine-tuning methods to the Yi(Young et al., 2024) and Qwen1.5(Bai et al., 2024) models with supervised instruction fine-tuning. To further enhance model performance, we incorporated multi-task learning methods and augmented the training set with synthetic data generated by LLMs. Finally, we developed a model ensemble approach involving re-inference for Hierarchical rhetorical classification tasks. Our method achieved a final score of 66.29 in CCL 2024 Eval Task 6 CERRU, ranking 1st.

2 Chinese Essay Rhetoric Recognition and Understanding

CCL 2024 Eval Task 6: Chinese Essay Rhetoric Recognition and Understanding includes three tracks:

Track 1: This track classifies rhetorical devices in each sentence at a coarse-grained level into five categories: metaphor, simile, hyperbole, parallelism, and no rhetoric. Additionally, each rhetorical category is further classified into subcategories based on form, resulting in a total of 4 coarse-grained categories and 12 fine-grained subcategories.

Track 2: Similar to Track 1, this track classifies rhetorical devices in each sentence at a coarse-grained level into the same five categories. However, the fine-grained classification is based on content, resulting in 4 coarse-grained categories and 11 fine-grained subcategories.

Track 3: This track focuses on identifying rhetorical components within sentences, specifically conjunction, tenor, and vehicle.

Overall, Tracks 1 and 2 fall under hierarchical text classification tasks, while Track 3 is an entity extraction task.

3 Methodology

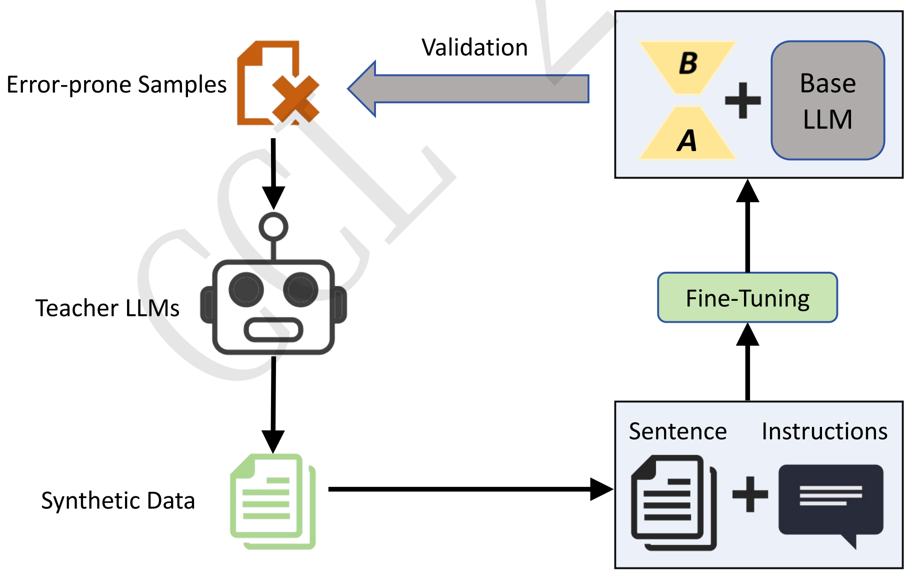

Figure 1: The process of generating synthetic data

3.1 Parameter-Efficient Instruction Fine-Tuning

Despite the evaluation task encompassing both classification and entity recognition, these tasks can be unified under an instruction-based format within the generative fine-tuning framework for LLMs. To fine-tune LLMs with limited hardware resources, we employed the LoRA (Hu et al., 2021) method for instruction fine-tuning of the base LLM.

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 223-231, Taiyuan, China, July 25 - 28, 2024.
Volume3: Evaluations

3.2 Multi-Task Learning

We believe that in the rhetorical classification task, not only should the LLM be informed of the rhetorical category to which a sentence belongs, but it should also identify the specific entities within the sentence that determine this category. This approach aids the LLM in better understanding and analyzing rhetorical categories within sentences. The same principle applies to the task of rhetorical entity extraction.

It is important to note that the datasets for the three tracks in the evaluation task contain identical text, differing only in their respective annotations. Therefore, implementing multi-task learning in our experiments was straightforward: we simply combined the instruction datasets from the three tracks and performed fine-tuning on this mixed dataset.

3.3 Synthetic Data

The limited number of annotated training samples provided for the evaluation task constrained further improvements in model performance. Inspired by the LLM2LLM method (Lee et al., 2024), we recorded error-prone samples in track 1&2 from the validation set during the fine-tuning process. As shown in Figure 1, we then used a more powerful LLM as a teacher model to generate synthetic data based on these error-prone samples.

3.4 Model Ensemble Based on Label Re-Inference

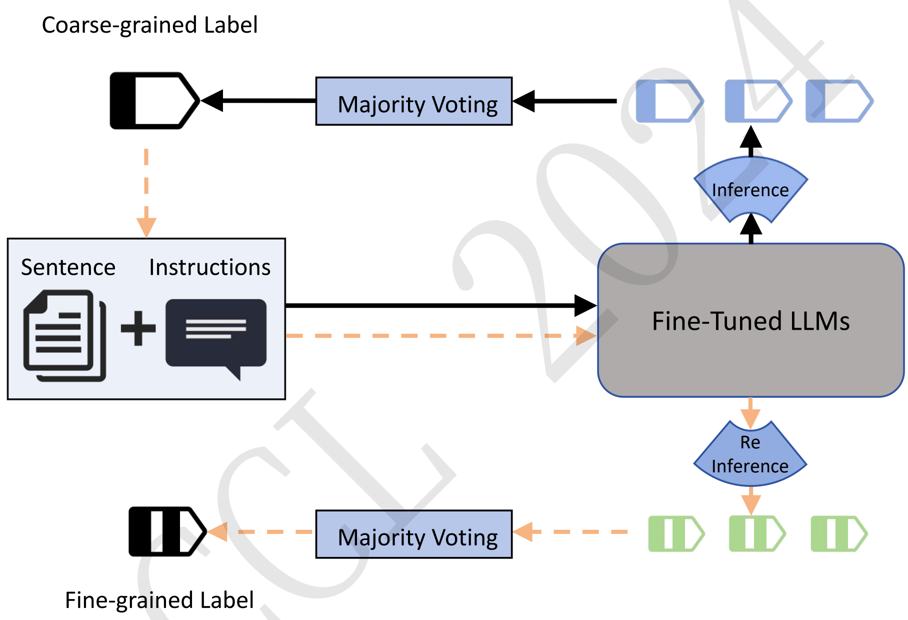

Figure 2: Model ensemble based on label re-inference

To further enhance model performance on track 1&2, we explored a model ensemble approach for the task of classifying coarse-grained and fine-grained labels using LLMs. This process is shown in Figure 2. Assume there are K fine-tuned LLMs, with the parameters of the k-th LLM denoted as θ_k. For a given sentence x, we add instructions to form $Instruction(x)$, which is then input into θ_k for inference, yielding a coarse-grained label $y_k^{(h)}$(**H**igh-level label) and a fine-grained label $y_k^{(l)}$(**L**ow-level label), along with their respective possible label sets $\mathcal{C}^{(h)}$ and $\mathcal{C}^{(l)}$:

$$(y_k^{(h)}, y_k^{(l)}) = \underset{(c^{(h)},c^{(l)})\in\mathcal{C}^{(h)}\times\mathcal{C}^{(l)}}{\arg\max} P\left(y_k^{(h)} = c^{(h)}, y_k^{(l)} = c^{(l)} \mid Instruction(x)\ , \theta_k\right) \tag{1}$$

Next, we perform majority voting on these coarse-grained label results:

$$y_{ens}^{(h)} = \arg\max \left(\sum_{k=1}^{K} \delta(y_k^{(h)} = c_j^{(h)})\right) \tag{2}$$

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 223-231, Taiyuan, China, July 25 - 28, 2024.
Volume3: Evaluations

where $j = \{1, 2, ..., |\mathcal{C}^{(h)}|\}$, and δ is the indicator function, which equals 1 when $y_k^{(h)} = c_j^{(h)}$ and 0 otherwise. Given that generative LLMs predict the next token based on the given sequence, and that the LLM has learned the constraints between coarse-grained and fine-grained labels during the fine-tuning phase, there is no need for additional fine-tuning. Provided $y_{ens}^{(h)}$ is not null, we append $y_{ens}^{(h)}$ to the instruction and re-input it into the LLM for re-inference under the constraint of the coarse-grained label:

$$y_k^{(l\_re)} = \underset{c^{(l)} \in \mathcal{C}^{(l)}(y_{ens}^{(h)})}{\arg\max} \; P\left(y_k^{(l\_re)} = c^{(l)} \mid Instruction(x) + y_{ens}^{(h)} \; , \theta_k\right) \tag{3}$$

We then conduct majority voting on the re-inferred fine-grained labels:

$$y_{ens}^{(l)} = \arg\max \left(\sum_{k=1}^{K} \delta(y_k^{(l\_re)} = c_j^{(l)}) \right) \tag{4}$$

where $j = \{1, 2, ..., |\mathcal{C}^{(l)}|\}$. The resulting $(y_{ens}^{(h)}, y_{ens}^{(l)})$ constitutes the final classification result for the sentence x.

4 Experiments

4.1 Dataset

The experimental dataset is derived from CCL 2024 Task 6 and includes three tracks. Each track consists of 634 samples in the training set, 225 samples in the validation set, and 5000 samples in the test set.

We processed the datasets into instruction formats, examples of the fine-tuning data can be found in the appendix. For training samples with multiple answers, we separated them using "\n".

4.2 Models

In our experiments, we used several LLMs for instruction fine-tuning: Yi-6B-Base, Qwen1.5-7B-Base, Qwen1.5-14B-Base, and Qwen1.5-32B-Base. We observed that, in most cases, larger LLMs tend to achieve better performance.

For generating synthetic data, we employed Qwen-max-0403 and Qwen-max-0428 as teacher models, accessed via API. The prompt used for calling the teacher models can be found in the appendix.

4.3 Settings

We used the AdamW optimizer with β_1 and β_2 set to (0.9, 0.999) and a weight decay of 1e-2. The initial learning rate was 3e-5, with cosine annealing applied to decay the learning rate to 6e-6 after 600 steps. The batch size was 8, and gradient checkpointing was enabled. The gradient norm was clipped to a maximum of 1.0.

For the LoRA hyperparameters, we used $r = 72$ and $\alpha = 612$, which is equivalent to $\alpha = 72$ with rsLoRA enabled. As noted in (Kalajdzievski, 2023), the scaling factor $\frac{\alpha}{r}$ in the original LoRA implementation (Hu et al., 2021) is too aggressive, leading to gradient collapse issues when using larger LoRA ranks. Adjusting the scaling factor to $\frac{\alpha}{\sqrt{r}}$ can mitigate this problem. Therefore, we adopted a larger LoRA α while using the original LoRA scaling factor.

During training, evaluation was performed on the validation set every 5 steps. We used the evaluation script provided by the organizers to compute the F1 score and saved the checkpoint with the highest score.

For inference, we employed beam search with (temperature=0.5, num_beams=3) to predict the target sequence.

The experiments were conducted on 1∼5 Nvidia A30 GPUs. The frameworks used for the experiments were Pytorch and HuggingFace Transformers.

Model	track1		track2		track3	
	val	test	val	test	val	test
Yi-6B-Base	49.08	53.72	52.12	48.63	67.11	68.00
Qwen1.5-7B-Base	49.83	56.29	52.59	53.63	68.35	-
Qwen1.5-14B-Base	**52.06**	**56.68**	**57.21**	**60.58**	69.98	70.97
Qwen1.5-32B-Base	51.66	-	55.18	-	**73.17**	**74.20**

Table 1: Model comparison

4.4 Model Performance Comparison

We tested several LLMs, fine-tuning them on the training set for each track. The results are presented in Table 1, where ”val” indicates the validation set score and ”test” indicates the score obtained after submitting the test set results.

Overall, larger models generally performed better. However, we observed that Qwen1.5-32B-Base did not outperform the smaller 14B model on the validation sets for Track 1&2. Ultimately, we selected Qwen1.5-14B-Base for subsequent experiments in Track 1 and Track 2, and Qwen1.5-32B-Base for the task in Track 3.

4.5 Evaluation Results

Table 2 shows the test set scores under different methods.

Track	Model&Method	Test Score
track1	Qwen1.5-14B-Base	56.68
	+MultiTaskLearning+ModelEnsemble	59.12
	+MultiTaskLearning+ModelEnsemble+SyntheticData+ValData	**61.30**
track2	Qwen1.5-14B-Base	60.58
	+MultiTaskLearning+ModelEnsemble	61.72
	+MultiTaskLearning+ModelEnsemble+SyntheticData+ValData	**62.29**
track3	Qwen1.5-32B-Base	74.20
	+MultiTaskLearning	74.61
	+MultiTaskLearning+PostProcessing	**75.28**

Table 2: Evaluation results

For Tracks 1&2, scores improved after applying multi-task learning and label re-inference model ensemble methods, with a more significant improvement in Track 1.

Subsequently, we used Qwenmax-0403 and Qwen-max-0428 as teacher models and generated synthetic data using error-prone samples from the original validation set as seeds. During this process, we found that the teachers provided highly repetitive answers for the same error-prone samples. Therefore, we manually filtered out the highly repetitive synthetic samples. Additionally, to ensure that the synthetic data was similar in length to the original samples, we removed synthetic samples that significantly deviated in length from the original samples based on their length ratio. This resulted in nearly 200 synthetic data samples.

Moreover, it is generally accepted that the validation set, being most similar in distribution to the original training set, is a crucial resource. We ultimately combined the validation set with 100 synthetic samples into the training set, leaving the remaining synthetic data as the validation set. This approach further improved our scores, reaching 61.30 for Track 1 and 62.29 for Track 2.

For Track 3, multi-task learning provided a slight score improvement.

We observed that the task definition for ”exaggeration” rhetoric did not include conjunctions. However, the model often included conjunctions in the prediction results. Therefore, we implemented a simple post-processing step: if a sentence was identified as exaggeration rhetoric in both Track 1 and Track 2, we removed the conjunctions from the Track 3 prediction. Similarly, if a sentence was labeled as ”no rhetoric” in both Track 1&2, we also set it to ”no rhetoric” in the Track 3 prediction. This final post-processing step increased the score to 75.28.

5 Conclusion

In this paper, we explored the use of large language models (LLMs) for the tasks of essay rhetoric recognition and understanding. By employing instruction fine-tuning, multi-task learning, synthetic data augmentation, and a model ensemble method based on label re-inference, we significantly improved the model's performance. Ultimately, our approach achieved a score of 66.29 in Task 6 of the CCL 2024 evaluation, CERRU, securing the first place. This outcome demonstrates the tremendous potential of LLMs in the automated processing of complex natural language tasks.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2024. Introducing qwen1.5.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Xiao Hu, Wei Song, Lizhen Liu, Xinlei Zhao, and Chao Du. 2017. Automatic recognition of simile based on sequential model. In *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, pages 410–413.

Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Sehrish Iqbal, Mladen Rakovic, Guanliang Chen, Tongguang Li, Rafael Ferreira Mello, Yizhou Fan, Giuseppe Fiorentino, Naif Radi Aljohani, and Dragan Gasevic. 2023. Towards automated analysis of rhetorical categories in students essay writings using bloom's taxonomy. pages 418–429. Association for Computing Machinery (ACM).

Damjan Kalajdzievski. 2023. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*.

Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipali, Michael W Mahoney, Kurt Keutzer, and Amir Gholami. 2024. Llm2llm: Boosting llms with novel iterative data enhancement. *arXiv preprint arXiv:2403.15042*.

Chunhong Li and Yongquan Li. 2022. A new method of figurative rhetoric recognition based on automated essay scoring of the oversea chinese students' instructional composition corpus. In *Proceedings of the 6th International Conference on Digital Technology in Education*, pages 107–111.

Lizhen Liu, Xiao Hu, Wei Song, Ruiji Fu, Ting Liu, and Guoping Hu. 2018. Neural multitask learning for simile recognition. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1543–1553, Brussels, Belgium, October-November. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. *arXiv preprint arXiv:2305.08377*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. 2023. Instructuie: multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 223-231, Taiyuan, China, July 25 - 28, 2024.
Volume3: Evaluations

Huang Xiaoxi, Li Hanyu, Wang Rongbo, Wang Xiaohua, and Chen Zhiqun. 2018. Recognizing metaphor with convolution neural network and svm. *Data Analysis and Knowledge Discovery*, 2(10):77–83.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Appendix

A Examples of the fine-tuning data

A.1 track1:

```
{
"instruction": "请你识别出以下句子中的修辞类别，这是一个层次多标签问题，共有4大类和12小类。若你认为有多个结果，请使用换行符隔开。\n句子：他变的行尸走肉,成天郁郁寡欢。\n选项：比喻：明喻，暗喻，借喻。比拟：名词，动词，形容词，副词。夸张：直接夸张，间接夸张，融合夸张。排比：成分排比，句子排比。",
"output": "夸张-直接夸张"
}
```

A.2 track2:

```
{
"instruction": "请你识别出以下句子中的修辞内容类型，这是一个层次多标签问题，共4大类11小类：他变的行尸走肉,成天郁郁寡欢。\n选项：比喻：实在物，动作，抽象概念。比拟：拟人，拟物。夸张：扩大夸张，缩小夸张，超前夸张。排比：并列，承接，递进。",
"output": "夸张-扩大夸张"
}
```

A.3 track3:

```
{
"instruction": "抽取出下列学生作文文本中的修辞成分。抽取结果格式为：\"连接词：xxx|描写对象：xxx|描写内容：xxx\"，若有多个结果，请使用换行符隔开。\n（1）针对比喻修辞，对于明喻形式，修辞成分包括连接词（喻词）、描写对象（本体）和描写内容（喻体）；对于暗喻形式，修辞成分包括描写对象（本体）和描写内容（喻体）；对于借喻形式，修辞成分包括描写内容（喻体）；\n（2）针对比拟修辞，不论形式如何，修辞成分都包括描写对象（比拟对象）和描写内容（比拟内容）；\n（3）针对夸张修辞，不论形式如何，修辞成分都包括描写对象（夸张对象）和描写内容（夸张内容）；\n（4）针对排比修辞，不论形式如何，修辞成分都包括连接词（排比项或排比标记）。\n文本：他变的行尸走肉,成天郁郁寡欢。\n抽取结果：",
"output": "描写对象：他|连接词：无|描写内容：变的行尸走肉"
}
```

B System prompt for calling the teacher LLMs

```
system_prompt = '''现在你是一个经验丰富的语文知识助手。
有一道题目是：以句子作为基本单位，将每个句子中使用的修辞手法按粗粒度分类成比喻、比拟、夸张、排比以及无修辞类，同时每类修辞进一步从形式角度细粒度分类。
所有类别和解释如下：
一、比喻：
从形式划分：
```

（1）明喻：本体、喻体、喻词都出现，通过使用像\像"\好像"\如"\仿佛"等比喻词来连接本体和喻体。如：仙人掌像几个绿色的手掌，有大有小，有粗有细。
（2）暗喻：仅本体、喻体出现，而没有\像"\好像"\如"\仿佛"这些比喻词。如：若要想要种子茁壮成长，少不了机会这一养料。
（3）借喻：仅喻体出现。如：一把弯刀挂在天上。
从内容划分：
（1）实在物：本体是可见、可触、可想的实在物体。如：一把弯刀挂在天上。
（2）动作：本体是某种动作、行为或事件。如：丁尽粳冬的一夜雨声，敲起了春耕的锣，擂响了播种的鼓。
（3）抽象概念：本体是抽象概念，如爱、时间、勇气等。如：时间就是金钱。
二、比拟：
从形式划分：
（1）名词：用写人的名词写物/用写物的动词写人或其他物。如：我看到这辆车久历风尘，实在高寿。
（2）动词：用写人的动词写物/用写物的动词写人或其他物。如：杜鹃花在风中摇曳，向人们展示它优美的舞姿。
（3）形容词：用写人的形容词写物/用写物的形容词写人或其他物。如：湖水愈发温柔，愈发安详。
（3）副词：用写人的副词写物/用写物的副词写人或其他物。如：高粱红了脸，羞答答地低下头微笑。
从内容划分：
（1）拟人：把非人当作人写。如：湖水愈发温柔，愈发安详。
（2）拟物：把非A的某物当作 A写，A非人。如：我到了自家的房外，我的母亲早已迎着出来了，接着便飞出了八岁的侄儿宏儿。
三、夸张：
从形式划分：
（1）直接夸张：直接对事物进行夸张。如：我的爱比所有加起来还要多。
（2）间接夸张：夸大另一样东西来夸大某事物。如：一海洋的水都洗不干净他的手。
（3）融合夸张：借助其他修辞进行夸张，通常是比喻。如：馋虫上身的我也管不了那么多了，拿着这钱就去了小商铺。
从内容划分：
（1）扩大夸张：向大、多、长或高等夸张。如：长长的队伍没有尽头，链接着五湖四海，千山万岭。
（2）缩小夸张：向小、少、短或低等夸张。如：随便你什么时候仰头看，只能看见巴掌大的一块天。
（3）超前夸张：把后出现的事说到先出现的事之前。如：还没回家就已经闻到香味了。
四、排比：
从形式划分：
（1）成分排比：指一个句子中的某些成分，如主谓宾定状等，通过重复的形式排列在一起。如：她的枕，她的床，她的房间，已经空了。
（2）句子排比：排比项可单独成句。如：它坚强的意志让我感动，它从不服从命运的安排，它与命运斗争，它触动了我的心灵。
从内容划分：
（1）并列：排比项顺序改变不影响语义通顺。如：工地上没人唱歌，没人跳舞，没人摔跤，没人吹牛皮，没人闹哄哄地赌饭吗。。。
（2）承接：排比项之间有先后逻辑顺序，如时间、程度、发展状况等，不能改变顺序。如：如果我没记错的话，他不就是三岁扎小辫、五岁穿花裤、九岁还吃奶的那个密级生么？
（3）递进：各排比项表达的含义、情感等层层递进，不能改变顺序。如：我与她深一脚浅一脚重新往黑暗里，往天塌地陷的前面闯，往一个几乎毫无希望的绝境里闯。
五、无修辞。

你发现有一些同学分不清一些句子所属的类别，需要更多的例子。现在需要你的帮助。我会给出一个例句，你需要参考例句，创造一个与其粗细类别都相同的新句子。
要求：
1.你给出的答案的类别只限定在上面给出的解释中。
2.确保不要出现错误。
3.确保新句子的类别与例句一致。若例句给出的答案有多个，你也需要保证新句子的答案个数一致，每个答案逗号隔开。
4.若例句中没有用到修辞，你需要给出一个没有用到任何修辞手法的句子。
5.创造与例句意义相似的句子，但不要复制原文
6.尽量使你给出的句子贴近学生真实写作风格。
7.若例句中为单一类别，请你尽量保证给出的新句子只包含与例句相同的修辞手法，尽量避免混入额外的修辞手法。
8.新句子长度尽量与例句相近。
9.发挥你的创造力。

一定要保证输出格式为如下格式：
<句子>：[句子]
<粗类别>：[粗类别]
<形式>：[形式]
<内容>：[内容]
'''

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 223-231, Taiyuan, China, July 25 - 28, 2024.
Volume3: Evaluations