System Report for CCL24-Eval Task 6: A Unified Multi-Task Learning Model for Chinese Essay Rhetoric Recognition and Component Extraction

Qin Fang, Zheng Zhang, Yifan Wang, Xian Peng*
National Engineering Research Center of Educational Big Data,
Central China Normal University, Wuhan, China
{fangqin, zhangz, wangyifan0122}@mails.ccnu.edu.cn
pengxian@ccnu.edu.cn

Abstract

In this paper, we present our system at CCL24-Eval Task 6: Chinese Essay Rhetoric Recognition and Understanding (CERRU). The CERRU task aims to identify and understand the use of rhetoric in student writing. The evaluation set three tracks to examine the recognition of rhetorical form, rhetorical content, and the extract of rhetorical components. Considering the potential correlation among the track tasks, we employ the unified multi-task learning architecture to fully incorporate the inherent interactions among the related tasks to improve the overall performance and to complete the above 3 track tasks with a single model. Specifically, the framework mainly consists of four sub-tasks: rhetorical device recognition, rhetorical form recognition, rhetorical content recognition, and rhetorical component extraction. The first three tasks are regarded as multi-label classification tasks, and the last task is regarded as an entity recognition task. The four tasks leverage potential information transfer to achieve fusion learning. Finally, the above four sub-tasks are integrated into a unified model through parameter sharing. In the final evaluation results, our system ranked fourth with a total score of 60.14, verifying the effectiveness of our approach.

Keywords: Multi-task learning , Rhetoric Recognition , Text Classification , Entity Recognition

1 Introduction

In the learning process of primary and secondary school students, rhetorical devices are not only a core component of reading comprehension and writing skills but also an indispensable element in shaping excellent literary works. Identifying and understanding the use of rhetoric in students' compositions can significantly enhance their expressive skills and guide them in producing higher-quality narratives and descriptions.

The CERRU task systematically defines the fine-grained rhetorical types found in primary and secondary school compositions, as detailed in Table 1. Evaluation in this task requires participating teams to not only identify rhetorical devices within sentences but also to conduct fine-grained classification of rhetorical form and content, and to provide the object and content of each rhetorical description. To achieve this, three tracks were established: rhetorical form recognition, rhetorical content recognition, and rhetorical component extraction. The task provided 634 training set examples, 225 validation set examples, and 5,000 test set examples.

Given the small number of training samples and the potential correlation between the track tasks, we have adopted a unified multi-task learning architecture to fully incorporate the inherent interactions among these related tasks, aiming to enhance learning efficiency and prediction accuracy (Zhang and Yang, 2021). By combining all tasks into a single model, we reduce computation and enable simultaneous completion of the three track tasks (Chen et al., 2021). Specifically, our framework consists of four

*Corresponding author

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 210-216, Taiyuan, China, July 25 - 28, 2024.
Volume3: Evaluations
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

修辞手法 Rhetorical Device	修辞形式类型 Form Type	修辞内容类型 Content Type
比喻 Simile	明喻, 暗喻, 借喻 Explicit, Implicit, Borrowed	实在物, 动作, 抽象概念 Concrete Objects, Actions, Abstract Concepts
比拟 Analogy	名词, 动词, 形容词, 副词 Noun, Verb, Adjective, Adverb	拟人, 拟物 Personification, Objectification
夸张 Hyperbole	直接夸张, 间接夸张, 融合夸张 Direct, Indirect, Integrated	扩大夸张, 缩小夸张, 超前夸张 Amplification, Diminishment, Anticipatory
排比 Parallelism	成分排比, 句子排比 Component, Sentence	并列, 承接, 递进 Coordination, Succession, Gradation

Table 1: The fine-grained rhetoric types in form and content

main sub-tasks: rhetorical device recognition, rhetorical form recognition, rhetorical content recognition, and rhetorical component extraction. The first three tasks are treated as multi-label classification tasks, while the last task is handled as an entity recognition task. Initially, we employ a transformer-based pre-trained language model as a shared feature encoder to represent sentences. Subsequently, the four tasks leverage potential information transfer to achieve fusion learning. Finally, these sub-tasks are integrated into a unified model through parameter sharing.

Additionally, we experimented with five different mainstream transformer-based pre-trained language models as backbone networks to assess their performance on the task. Given that multi-task learning requires optimizing models for multiple objectives, we also experimented with four different loss weighting schemes to approach the optimal performance of the model.

In this paper, our contributions can be summarized in three main aspects:

(1) We propose a multi-task learning framework that integrates related subtasks, enhancing interactions between them. This approach allows us to use a single model to complete all three track tasks effectively.

(2) We compare the performance of five different pre-trained language models as backbone networks and explore four weighting methods to optimize the model's performance.

(3) Our proposed framework achieved fourth place in the CCL24-Eval Task 6 (Chinese Essay Rhetoric Recognition and Understanding, CERRU) with a total score of 60.14, demonstrating the effectiveness of our method.

2 Methodology

To fully leverage the potential correlation between each task, we employ a multi-task learning framework. This approach can be seen as an inductive knowledge transfer method that improves generalization by sharing domain information across complementary tasks. By learning multiple tasks using shared representations, insights gained from one task can aid in learning the others (Caruana, 1997). Additionally, to further enhance the model's generalization ability, we incorporate adversarial training methods during model training.

2.1 Model Architecture

Figure 1 illustrates an overview of the framework. During the training phase, each task has its corresponding objective function, and all task-specific training data are used to jointly train the model in a bottom-up order.

Shared Feature Encoder The shared feature encoder focuses on mapping the input tokens to distributed semantic representations, which are shared across four downstream subtasks. To better capture and summarize the semantics of a given sentence, we adopt a pre-trained language model based on Transformer (Vaswani et al., 2017) as the shared feature encoder and fine-tune it based on the joint loss function of multi-task learning.

Given an input sentence $X = \{x_1, x_2, \cdots, x_n\}$, we first insert special tokens $[CLS]$ at the beginning and

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 210-216, Taiyuan, China, July 25 - 28, 2024.
Volume3: Evaluations

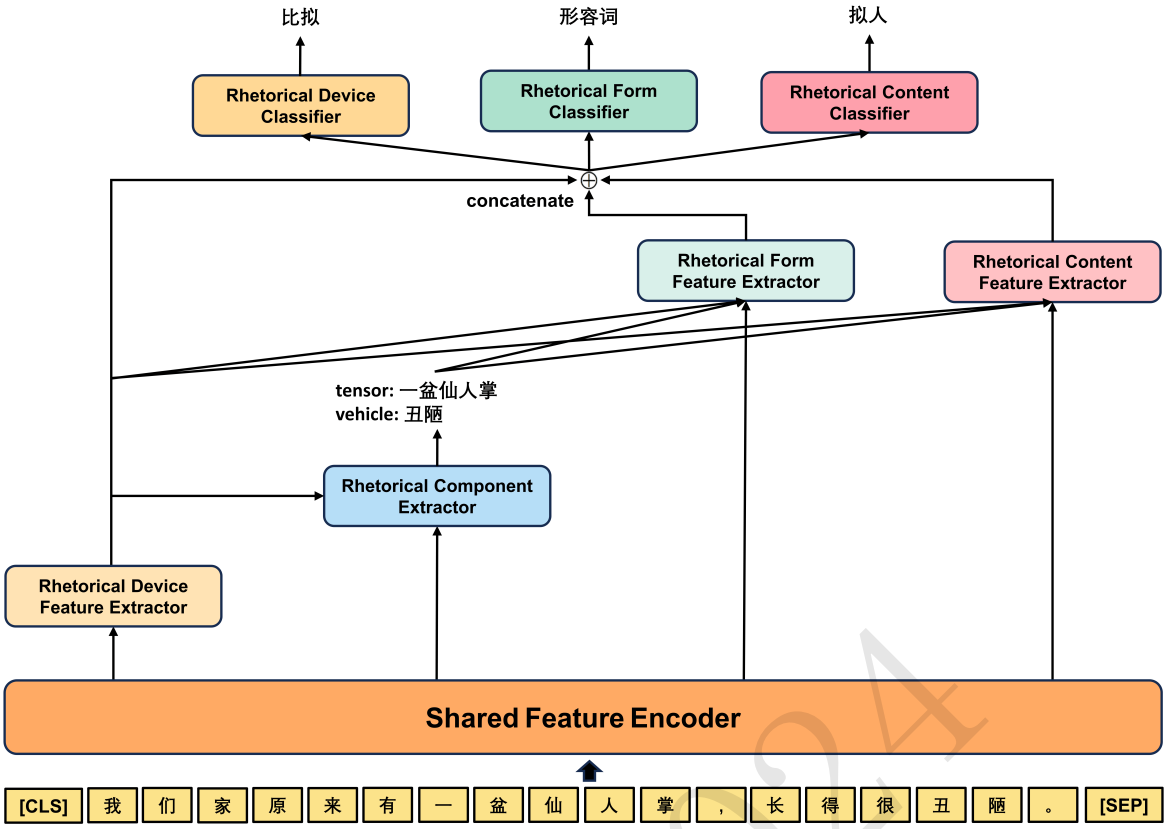

Figure 1: Model Architecture

$[SEP]$ at the end. The processed sequence is then input into the shared feature encoder. Subsequently, the shared feature encoder generates semantic representations for each token, with the output represented as $O^{encoder}$.

Rhetorical Device Feature Extractor Due to the use of an improved BERT-based pre-trained model in the shared feature encoder, which captures rich contextual information through Bidirectional Encoder Representations (BERT) (Devlin et al., 2018), we further utilize a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) network to enhance sentence representations and extract comprehensive features of rhetorical devices. BiLSTM leverages both forward and backward LSTM directions for feature extraction, capturing semantic features across contexts to obtain more comprehensive feature information. The final feature output of the rhetorical device feature extractor is denoted as F^{device}.

Rhetorical Component Extractor To enhance the extraction of rhetorical components from sentences, we concatenate rhetorical device features with semantic representations. Specifically, F^{device} is simply appended at the beginning of $O^{encoder}$, resulting in $[F^{device}, O^{encoder}]$. Although other concatenation methods were not considered, this straightforward approach effectively integrates the potential information from rhetorical device features, enhancing the module's performance. Subsequent experimental results have validated the effectiveness of this method.

For accurate entity boundary identification, we employ Efficient GlobalPointer (Su et al., 2022), a span-based entity recognition method. Efficient GlobalPointer uses two modules to detect the start and end positions of entities within a sentence, allowing for the classification of sentence subsequences as named entities. Figure 2 illustrates a matrix corresponding to two types of entities in the sentence. Compared to GlobalPointer, Efficient GlobalPointer achieves comparable performance with fewer parameters.

Rhetorical Form and Content Feature Extractor The architectures of the rhetorical form extractor and the rhetorical content extractor are identical. The specific process begins with local context

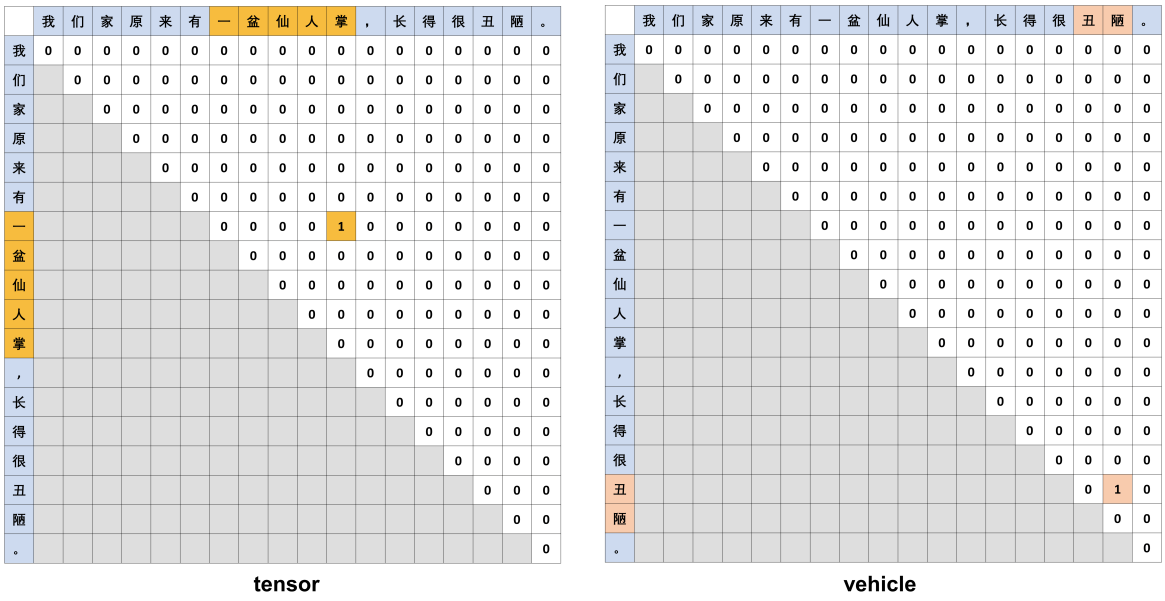

Figure 2: Entity recognition decoding structure based on GlobalPointer

enhancement on the contextualized word representations based on the entity information output by the rhetorical component extractor. Subsequently, rhetorical device features are concatenated to fuse with the enhanced representations. Feature extraction is then performed using multi-head self-attention and BiLSTM to capture comprehensive feature information. Finally, each feature extractor outputs F^{form}(for rhetorical form) and $F^{content}$(for rhetorical content).

Classifier The rhetorical device classifier, rhetorical form classifier, and rhetorical content classifier share the same architecture. Recognizing the correlation between rhetorical devices, forms, and content, we concatenate the outputs of the three rhetorical feature extraction modules, denoted as $[F^{device}, F^{form}, F^{content}]$. Initially, these inputs are cross-fused and feature-extracted using BiLSTM. Subsequently, the classification results are produced through a fully connected network.

2.2 Adversarial Training

In the field of natural language processing, adversarial training is employed as a regularization method to improve a model's generalization performance. We incorporated FGM (Miyato et al., 2016) to add adversarial training to our model. FGM introduces an adversarial attack in the direction opposite to the gradient during backpropagation in the embedding layer, thereby inducing adversarial training effects. This training method not only enhances the model's generalization ability but also improves its robustness.

2.3 Loss Function

Overall, the input sentence X is encoded by the shared feature encoder. The contextual outputs are then used to compute four tasks with task-specific labels Y_i for i = 1, 2, 3, 4. We jointly optimize the integrated loss during training as follows:

$$\mathcal{L}_{tot}(\mathbf{X}, \mathbf{Y}_{1:4}) = \sum_{i=1}^{4} \lambda_i \mathcal{L}_i(\mathbf{X}, \mathbf{Y}_i) \tag{1}$$

where $\mathcal{L}_i$ represents the cross-entropy loss for each task, and λ_i is the weighting factors that balance the contribution of each task's loss to the overall loss. The overall model loss can be heavily influenced by one task due to the varying loss magnitudes across different tasks, causing other tasks to have less impact on the learning process of shared network layers. To mitigate this, it is crucial to choose appropriate

weights to balance the training of each task, ensuring all tasks contribute effectively to the model's improvement.

In our experiments, we used four weighting schemes to determine the weights suitable for our model. These include equal weighting(λ_i=1), weight uncertainty (Kendall et al., 2017), random loss weight (Lin et al., 2021) and dynamic weight average (Liu et al., 2019). Readers can refer to the earlier citations for implementation details of these methods.

3 Experiments

3.1 Experiment Setting

During training, we use the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 2e-5 and a batch size of 16. The maximum sequence length is set to 256, and the maximum number of epochs is 30. The random seed is set to 1018. Additionally, we employ a learning rate warm-up strategy where the number of warm-up steps is 10% of the total number of training steps.

3.2 Experimental Setup

For the shared feature encoder, we used five mainstream transformer-based pre-trained language models as the backbone network. These include Chinese-MacBERT-Large (Cui et al., 2020), Chinese-RoBERTa-WWM-Ext-Large (Cui et al., 2019), StructBERT-Large-Zh (Wang et al., 2019), Erlangshen-DeBERTa-v2-710M-Chinese (Zhang et al., 2022), and ERNIE-3.0-Xbase-Zh (Sun et al., 2021). For each backbone network, we experimented with four weighting methods as described in Section 2.3. Their performance on the validation set is shown in Table 2.

Backbone Network	Weighting Scheme	Score			
		Track 1	Track 2	Track 3	Total
Chinese-MacBERT-Large	Equal Weights	**48.08**	51.59	63.65	54.44
	Uncert. Weights	46.28	52.51	63.44	54.08
	Random Loss Weight	46.67	**53.85**	**63.73**	**54.75**
	Dynamic Weight Average	43.92	50.31	63.51	52.58
chinese-roberta-wwm-ext-large	Equal Weights	46.09	52.73	63.86	54.23
	Uncert. Weights	46.28	52.24	**65.35**	54.62
	Random Loss Weight	41.50	50.17	63.63	51.77
	Dynamic Weight Average	**46.34**	**53.98**	63.66	**54.66**
StructBERT-Large-Zh	Equal Weights	**49.44**	**55.10**	62.24	**55.59**
	Uncert. Weights	45.66	51.49	62.51	53.22
	Random Loss Weight	46.44	52.39	64.43	54.42
	Dynamic Weight Average	45.45	53.19	**64.50**	54.38
Erlangshen-DeBERTa-v2-710M	Equal Weights	43.65	50.99	66.90	53.85
	Uncert. Weights	42.30	48.53	66.09	52.31
	Random Loss Weight	**48.01**	**52.61**	67.58	**56.06**
	Dynamic Weight Average	41.63	50.01	**68.07**	53.23
ERNIE-3.0-Xbase-Zh	Equal Weights	43.95	53.17	66.36	54.50
	Uncert. Weights	45.95	**54.78**	**67.40**	**56.04**
	Random Loss Weight	**46.14**	53.27	66.70	55.37
	Dynamic Weight Average	45.98	54.15	66.83	55.65

Table 2: The performance of each backbone network with different weighting methods on the validation dataset. The best performing combination of backbone network and weighting is highlighted in bold. The top validation scores for each metric are annotated with boxes.

3.3 Experimental Results

We selected the optimal weighting method for each pre-trained model based on their performance on the validation set, identifying the top five performing models. Subsequently, during the last five epochs of training, we applied the Stochastic Weight Averaging (SWA) (Izmailov et al., 2018) method to these models for evaluation on the test set. For the final evaluation, we employed model ensemble voting. Our approach achieved a fourth-place ranking in the final evaluation results. The scores of the top five teams and baseline are presented in Table 3.

Team	Track 1	Track 2	Track 3	Score
Team1	61.30	62.29	75.28	66.29
Team2	59.20	60.92	77.96	66.03
Team3	53.77	60.15	68.26	60.72
Our team	50.86	55.81	73.75	60.14
Team5	51.48	55.11	69.51	58.70
Baseline	45.66	56.89	20.85	41.13

Table 3: The final evaluation results of the top five teams and Baseline

3.4 Results Analysis

Based on the evaluation results, our team's overall performance is close to the third position. Specifically, our performance across different tracks is as follows: on Track 3, our results significantly exceed the baseline (52.9 points) and the third-place score (5.49 points), whereas on Track 1, our performance is moderate, just slightly above the baseline (5.2 points), and on Track 2, our performance is below the baseline (1.08 points).

The experimental results demonstrate that our adopted multi-task learning model architecture achieves better generalization capability through shared representations across tasks. In particular, the model shows significantly enhanced entity recognition capabilities on Track 3.

However, our performance in classification tasks on Track 1 and Track 2 is relatively average. This could be attributed to the multi-task learning model needing to concurrently optimize loss functions across multiple sub-tasks, a design that prevents multi-task learning from achieving the optimal performance on each sub-task as in single-task learning. Nevertheless, the advantage of multi-task learning lies in its ability to use fewer model parameters to accomplish learning and inference efficiently for multiple sub-tasks with a single model.

In conclusion, our experimental results highlight both the potential and limitations of multi-task learning in terms of cross-task generalization capability.

4 Conclusion

In this paper, we propose a unified multi-task learning framework for CCL24-Eval Task 6 (CERRU), to enhance feature fusion and interaction among subtasks, and achieve a single model capable of completing all subtasks. We experimented with various pre-trained language models and weighting methods, and further improved our experimental results through model voting. Our experiments demonstrate that our proposed approach achieves good results in this evaluation.

However, there are still several shortcomings in this system. In the future, we plan to enhance the model's performance through data augmentation and domain-specific pre-training. Additionally, we intend to explore more suitable weighting methods for this evaluation.

References

Caruana, R. (1997). Multitask learning. *Machine learning*, 28:41–75.

Chen, S., Zhang, Y., and Yang, Q. (2021). Multi-task learning in natural language processing: An overview. *ACM Computing Surveys*.

Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., and Hu, G. (2020). Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z., Wang, S., and Hu, G. (2019). Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2018). Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence*.

Kendall, A., Gal, Y., and Cipolla, R. (2017). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7482–7491.

Lin, B., Ye, F., Zhang, Y., and Tsang, I. W.-H. (2021). Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Trans. Mach. Learn. Res.*, 2022.

Liu, S., Johns, E., and Davison, A. J. (2019). End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880.

Loshchilov, I. and Hutter, F. (2017). Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101.

Miyato, T., Dai, A. M., and Goodfellow, I. (2016). Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Su, J., Murtadha, A., Pan, S., Hou, J., Sun, J., Huang, W., Wen, B., and Liu, Y. (2022). Global pointer: Novel efficient span-based approach for named entity recognition. *ArXiv*, abs/2208.03054.

Sun, Y., Wang, S., Feng, S., Ding, S., Pang, C., Shang, J., Liu, J., Chen, X., Zhao, Y., Lu, Y., et al. (2021). Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, W., Bi, B., Yan, M., Wu, C., Bao, Z., Xia, J., Peng, L., and Si, L. (2019). Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*.

Zhang, J., Gan, R., Wang, J., Zhang, Y., Zhang, L., Yang, P., Gao, X., Wu, Z., Dong, X., He, J., Zhuo, J., Yang, Q., Huang, Y., Li, X., Wu, Y., Lu, J., Zhu, X., Chen, W., Han, T., Pan, K., Wang, R., Wang, H., Wu, X., Zeng, Z., and Chen, C. (2022). Fengshenbang 1.0: Being the foundation of chinese cognitive intelligence. *CoRR*, abs/2209.02970.

Zhang, Y. and Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609.