System Report for CCL24-Eval Task 1: Application of Entity Classification Model Based on Different Position Embedding in Chinese Frame Semantic Parsing

Huirong Zhou, Sujie Tian, Junbo Li, Xiao Yuan
School of Statistics, Beijing Normal University
{202322011204,202322011168,202322011137,202322011194}@mail.bnu.edu.cn

Abstract

This paper addresses three subtasks of Chinese Frame Semantic Parsing based on the BERT and RoBERTa pre-trained models: Frame Identification, Argument Identification, and Role Identification. In the Frame Identification task, we utilize the BERT PLM with Rotary Positional Encoding for the semantic frame classification task. For the Argument Identification task, we employ the RoBERTa PLM with T5 position encoding for extraction tasks. In the Role Identification task, we use the RoBERTa PLM with ALiBi position encoding for the classification task. Ultimately, our approach achieved a score of 71.41 in the closed track of the B leaderboard, securing fourth place and validating the effectiveness of our method.

Keywords: Chinese Frame-Semantic Parsing Relative Position Encoding Multi-Target Words

1 Introduction

The Chinese FrameNet (CFN) is a Chinese lexical semantic knowledge base for computer use, based on Fillmore's Frame Semantics theory and referencing FrameNet from the University of California, Berkeley, with Chinese real corpus as the basis (Hao et al., 2007). Frame semantic parsing involves a deep understanding of the entities or meanings involved in a sentence, which is of great significance in downstream tasks such as text summarization, relation extraction and reading comprehension.

The basic structure of frame semantic parsing is as follows: as shown in Figure 1, in the example sentence "从海拔2800米的仁青岗村到海拔4600多米的詹娘舍哨所" ("From Renqinggang Village at an altitude of 2800 meters to Zhanniangshe Outpost at an altitude of over 4600 meters"), the phrase "从……到……" ("from …… to ……") can activate a "moving path" frame, where "海拔2800米" ("at an altitude of 2800 meters"), "仁青岗村" ("Renqinggang Village"), "海拔4600多米" ("at an altitude of over 4600 meters"), and "詹娘舍哨所" ("Zhanniangshe Outpost") are the four arguments of the example sentence. The attributes of the arguments are respectively "feature, starting point, feature, end point".

Chinese Frame Semantic Parsing is a semantic parsing task based on Chinese Frame Semantic resources. This task consists of the following three subtasks: frame identification, argument identification, and role identification. The frame identification task involves identifying the corresponding frame category from candidate frame categories based on the target word given in the sentence. The argument identification task determines the boundaries of the arguments in the frame based on the target word in the sentence. The role identification task determines the names of the frame elements corresponding to each argument based on the results of argument identification. Therefore, this paper categorizes the frame identification task and role identification task as classification problems, and the argument identification task as an extraction task.

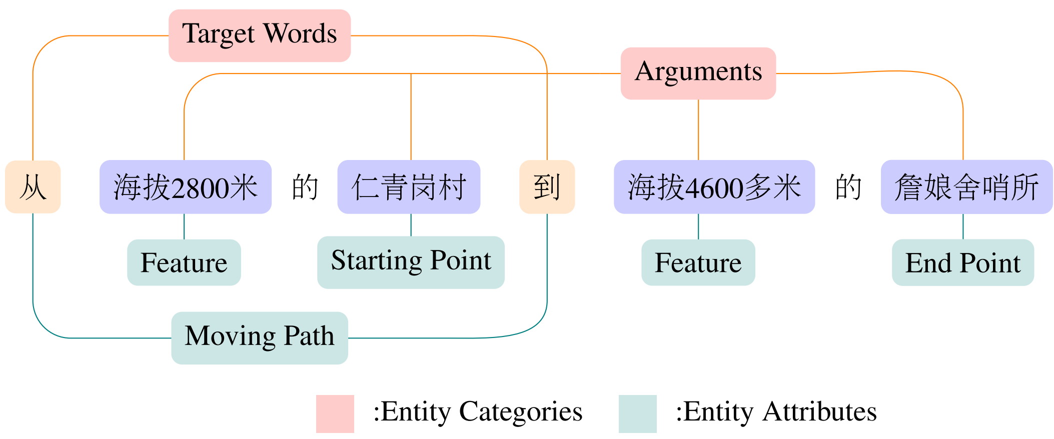

Figure 1: Training Data Examples

2 Related Work

In the task of Chinese Frame Semantic Parsing, traditional machine learning algorithms were initially used, such as maximum entropy (李济洪 et al., 2011), conditional random fields (李济洪, 2010), "OBI data type" annotation, and greedy algorithms (石佼 et al., 2014). However, with the continuous development of deep learning algorithms, Zhao et al. (赵红燕 et al., 2016) used DNN for frame identification, Suhaifeng et al. (Su et al., 2021) utilized frame relationships and definitions for frame identification, Zhou et al. (Zhou et al., 2021) identified frame elements and frames together in end-to-end tasks, and utilized the relationship between target words and frame elements to assist in frame identification during decoding. Wang et al. (王晓晖 et al., 2022) proposed a Chinese Frame Semantic Role Labeling method based on self-attention mechanism to obtain long-distance information from sentences.

The Chinese FrameNet (CFN) contains extensive resources and applications that are significant for natural language understanding tasks such as frame disambiguation and semantic role labeling. CFN mainly focus on common knowledge within contexts(Li et al., 2024), utilizing a combination of top-down, bottom-up, and expert manual curation methods for its creation. (Liu et al., 2023) used different end-to-end frameworks for parsing and employing data augmentation and voting methods to further improve prediction accuracy. (Li et al., 2023) propose an entity classification model based on Rotary Position Embedding (RoPE).(Huang et al., 2023) used a multi-task pipeline strategy and pre-trained language models to address the problem of Chinese Frame Semantic Parsing.

3 Methodology

3.1 Extraction Method for *Span* Type Data

In the task of *Argument Identification*, we need to "extract" entities from sentences. In addition to the token prediction extraction method for "OBI" data, our paper adopts an extraction method for a type of data called "span" data, where predictions are made for the start and end of the arguments. We treat each given sentence *"s"* as a "span" type data and label it with a "head-tail matrix". The head-tail matrix is an upper triangular matrix, and can be used as follows: the row number (vertical coordinate) represents the starting index of the predicted argument, while the column number (horizontal coordinate) represents the ending index of the predicted argument. "1" is marked at the start and end indices of the predicted argument, while "0" is marked in all other positions of the matrix.

As shown in Figure 2, the predicted arguments for the given sentence include "海拔2800米", "仁青岗村", "海拔4600多米",and "詹娘舍哨所". Taking "海拔2800米" as an example, the elements at the starting index (the row corresponding to "海") and the ending index (the column corresponding to "米") in the head-tail matrix should be marked as "1", and the pair of (starting index, ending index) should be treated as an extracted argument span.

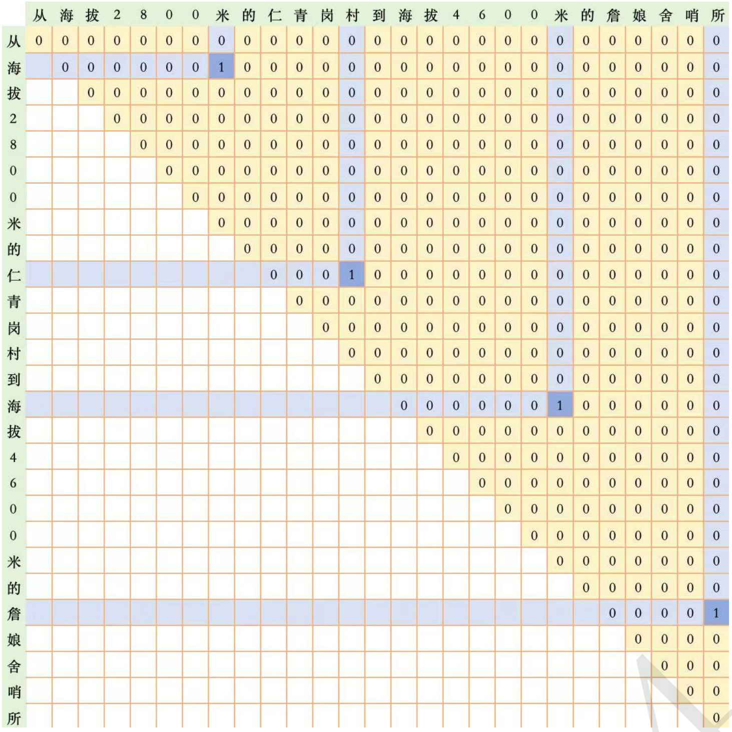

Figure 2: An example of using H-T matrix of "*span*" data

3.2 Method of Frame Identification Based on Multiple Target Words

The main task of *Frame Identification* is to map the semantic structure of a sentence onto a predefined semantic frame. Each frame represents a specific event, action, or situation and defines the roles associated with it, known as *frame elements*. For example, a "purchase" frame may include roles such as buyer, seller, product, and price.

The target word refers to the word or phrase in the sentence that triggers or activates a specific semantic frame. For instance, in the sentence "小明吃了一个苹果" ("Xiao Ming ate an apple"), the target word "吃" ("ate") activates the semantic frame of "eating". In more complex cases, there may be multiple target words in a given sentence. For example, in the sentence "从早上开始，小明一直在学习，直到傍晚才休息" ("Starting from the morning, Xiao Ming has been studying until taking a break in the evening"), the target phrase "从......到......" ("from......to......") activates the semantic frame of "time span". Unlike the activation of semantic frames with single target words, multiple target words often accompany complex grammatical structures, requiring techniques such as dependency parsing to accurately understand the relationships between sentence structures and components. In our paper, we attempt to use the mean or maximum attention scores of individual target words in the sentence as the predicted scores for the case of multiple target words.

3.3 Rotary Position Embedding(RoPE)

Traditional Transformers adopt either learned or sinusoidal absolute position encoding, lacking relative positional relationship information. However, Rotary Positional Encoding (RoPE) (Su et al., 2024) is an improvement upon the additive positional encoding usually applied in Transformers $(\mathbf{X} + \mathbf{P})$, which provides relative positional information between tokens. RoPE further assists the attention model in memorizing directional (preceding or following) information between tokens by employing multiplicative positional encoding $(\mathbf{X} \otimes \mathbf{P})$.

3.4 ALiBi Relative Position Encoding

Since the self-attention mechanism in $Transformer$ is independent of the text order, it is usually necessary to provide explicit positional signals to the $Transformer$. The original $Transformer$ uses sinusoidal or learned positional embeddings. Although absolute positional encoding is simple to implement and suitable for fixed-length sequences, it performs poorly when handling sequences of different lengths and capturing relative positional relationships. In contrast, relative positional encoding excels in capturing the relative relationships between elements and achieving translational invariance, making it more suitable for variable-length sequences. Recently the use of relative positional embeddings has become more common. Relative positional embeddings do not use fixed embeddings for each position but generate different learned embeddings based on the offsets between the key and $query$ being compared in the self-attention mechanism.

ALiBi *(Attention with Linear Biases)* positional encoding adds a linearly decreasing penalty proportional to the distance to the dot product of key and $query$ in the $Attention$ model.

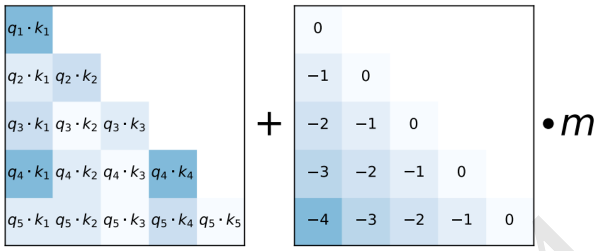

Figure 3: ALiBi

As shown in Figure 3, the left diagram is similar to the traditional $Transformer$, where the initial *attention score* is obtained through the dot product of key and $query$. The right diagram shows a relative distance matrix, where the elements of the matrix are the differences between the indices i and j of q_i and k_j. The third term m is a fixed slope parameter, which depends on the number of $heads$ in the $Attention$ (Press et al., 2021).

3.5 T5 Relative Position Encoding

The $T5$ relative position encoding is similar to the $ALiBi$ position encoding, where a penalty term is added to the inner product of the k and q in the corresponding *self-attention score matrix*. The difference is that this penalty term is not the same linearly decreasing penalty as in $ALiBi$. For example, let x_i and x_j represent the tokens at position i and j respectively. According to the original $Transformer$ principle formula (1), the inner product $e_{ij}^{(h)}$ of the k and q in $self-attention$ is computed, and then a penalty term r_{ij} is added according to formula (2) to obtain $\hat{e}_{ij}^{(h)}$, followed by a $softmax$ operation on $\hat{e}_{ij}^{(h)}$, as shown in formula (3). The relative position encoding r_{ij} is a scalar value. The $T5$ (Roberts et al., 2019) uses a partitioning method to map various relative position information into a total of 32 types. For example, the relative position information between nearby tokens is more important and thus needs to be more precise. Conversely, distant relative positions do not need to be as precise, so $T5$ divides these distant positions into different regions, using the same value within the same region, as shown in the mapping relationship in $Table$ 1.

$$e_{ij}^{(h)} = \frac{\boldsymbol{x_i} W_Q^{(h)} \left(\boldsymbol{x_j} W_K^{(h)}\right)^\top}{\sqrt{d_z/H}}, \tag{1}$$

$$\hat{e}_{ij}^{(h)} = e_{ij}^{(h)} + r_{ij}, \tag{2}$$

$$\alpha_{ij}^{(h)} = \operatorname*{softmax}_j \left\{\hat{e}_{ij}^{(h)}\right\}. \tag{3}$$

$i-j$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$r(i-j)$	0	1	2	3	4	5	6	7	8	8	8	8	9	9	9	9
$i-j$	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	$\cdots$
$r(i-j)$	10	10	10	10	10	10	10	11	11	11	11	11	11	11	11	$\cdots$

Table 1: T5 Relative Position Mapping

3.6 Warm-up Strategy

The warm-up learning strategy involves starting with a small learning rate at the beginning of training and then gradually increasing it to the preset learning rate. The purpose of this is to update the model parameters more stably in the early stages of training, preventing model instability or divergence caused by an excessively high initial learning rate. The linear warm-up strategy refers to starting the learning rate from a small value and linearly increasing it to the preset learning rate. The expression for updating the learning rate is as follows:

$$\mathrm{lr}_t = \mathrm{lr}_0 + \frac{(\mathrm{lr}_{\max} - \mathrm{lr}_0) \times t}{T}. \tag{4}$$

In this context, lr_t is the learning rate at the t-th iteration, lr_0 is the initial learning rate (usually set to a very small value), and $\mathrm{lr}_{\max}$ is the preset maximum learning rate (usually the final target learning rate). t is the current iteration number, and T is the total number of warm-up steps. According to the above formula, it is clear that within the first T steps of training, the learning rate increases linearly from the initial value lr_0 to the maximum value $\mathrm{lr}_{\max}$. After exceeding T steps, the learning rate further decreases until convergence.

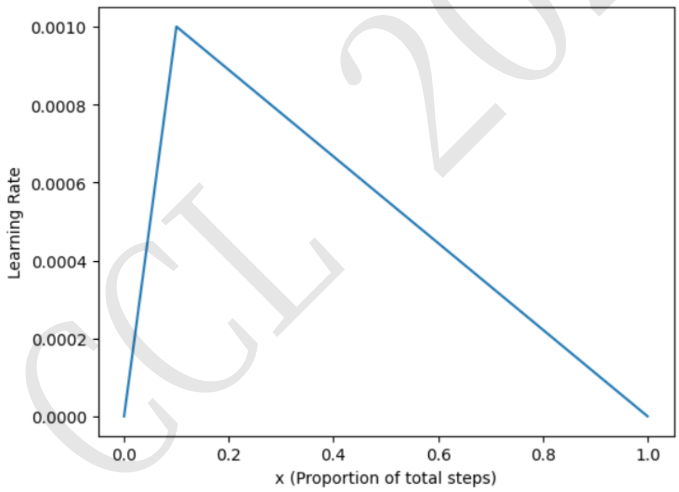

Figure 4: Warm-up Linear Strategy

Figure 4 shows the change in the learning rate during the training process using linear warm-up. In the example shown in the figure, $\mathrm{lr}_{\max} = 0.001$, $\mathrm{lr}_0 = 0$, and T accounts for 0.1 of the total training steps.

4 Model Description

4.1 Task 1: Frame Identification

According to the given sentence $s =< w_1, \cdots, w_i, \cdots, w_j, \cdots, w_n >$ of length n and the context of the target word t, we aim to find the most likely frame f_t to be activated from the frame set $F = \{f_1, f_2, \cdots, f_T\}$, which can be formulated as:

$$f_t = \underset{1 \leq j \leq T}{\operatorname{argmax}} P\left(f_j \mid s, t\right). \tag{5}$$

The probability $P(f_j \mid s, t)$ is estimated as follows: In the given sentence s, the target word t has the first and last characters w_{ti} and w_{tj}, respectively. We obtain word vectors $\mathbf{w}_i^f = \mathbf{E}(w_{ti}, s, f)$ and $\mathbf{w}_j^f = \mathbf{E}(w_{tj}, s, f)$ using word embedding methods, and then compute the relative information between the two word vectors $\mathbf{I}\left(\mathbf{w}_i^f, \mathbf{w}_j^f\right)$. For each frame f in the frame set F:

$$P(f \mid s,t) \approx \frac{\exp\left(\mathbf{I}\left(\mathbf{w}_i^f, \mathbf{w}_j^f\right)\right)}{\sum_{\hat{f}\in F} \exp\left(\mathbf{I}\left(\mathbf{w}_i^{\hat{f}}, \mathbf{w}_j^{\hat{f}}\right)\right)}. \tag{6}$$

The architecture of the frame identification task model is illustrated in Figure 5.

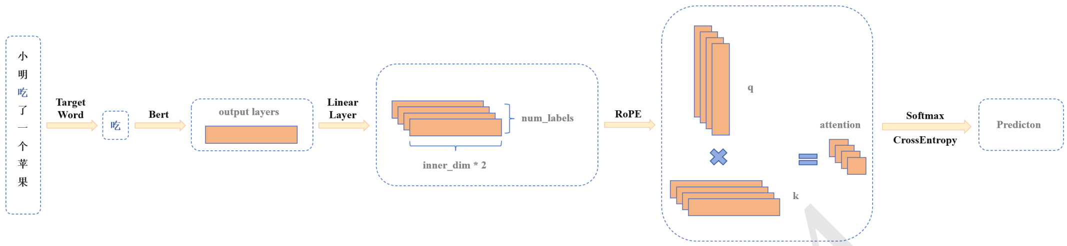

Figure 5: Frame Identification Model

In the model framework shown in Figure 5, for each frame f_t in the frame set F, given the boundaries of the target word w_i, w_j, a corresponding score can be computed as:

$$S_t = S_t\left(w_i^t, w_j^t\right). \tag{7}$$

The activated semantic frame should then be:

$$f = \underset{1\leq t\leq T}{argmax}\left(softmax\left(\mathbf{S}\left(\mathbf{w}_i, \mathbf{w}_j\right)\right)\right), \tag{8}$$

where:

$$\mathbf{S}\left(\mathbf{w}_i, \mathbf{w}_j\right) = [S_1(w_i^1, w_j^1), S_2(w_i^2, w_j^2), \cdots, S_T(w_i^T, w_j^T)]. \tag{9}$$

The loss function during model training is defined as:

$$CrossEntropyLoss = CrossEntropyLoss(\mathbf{S}\left(\mathbf{w_i}, \mathbf{w_j}\right), real\_frame\_label). \tag{10}$$

4.2 Task 2: Argument Identification

The main purpose of the argument identification task is to determine the positions of each argument (i.e., frame element) involved by each target word in the sentence. That is, given a sentence $s =< w_1, w_2, ..., w_n >$ and a target word t, the task is to find the set of arguments $\hat{T}$ matched with the target word in the set of all consecutive ordered word elements $T = \{t_1, t_2, ..., t_N\}$ within the sentence.

As the training samples provided contain the starting and ending indices of the arguments under the corresponding framework, this problem can be converted into a multi-classification task. Following the approach of $GlobalPointer$ (苏剑林, 2021), this paper adopts the extraction method of predicting the heads and tails of $span$-type data. For a sentence s of length n, there are a total of $n(n+1)/2$ candidate arguments. If the sentence s has k argument indices, then this extraction task is transformed into a multi-label classification problem of selecting k out of $n(n+1)/2$.

For a sentence s of length n, after encoding, it yields a vector sequence $[v_1, v_2, ..., v_n]$. Based on the $Transformer$ transformation, vector sequences $[q_1, q_2, ..., q_n]$ and $[k_1, k_2, ..., k_n]$ are obtained. Then

$RoPE$ (rotary positional encoding) is introduced, where a transformation matrix R_i satisfies $R_i^T R_j = R_{j-i}$. According to formula (11), rotation is applied to q and k to calculate the attention score $s(i,j)$ between the i-th and j-th word elements, explicitly incorporating relative positional information into the attention score. Meanwhile, for $Task2$, our paper introduces the $T5$ position encoding, adding a penalty term for relative position to the attention score, and obtains $logits$ according to formula (12). Based on the discriminant formula (13), the indices (i,j) where $logits(i,j) = 1$ are obtained, which represent the predicted argument scopes. Finally, our paper employs the loss function used in $GlobalPointer$ as shown in formula (14).

$$s(i,j) = (R_i q_i)^T (R_j k_j) = q_i^T R_i^T R_j k_j = q_i^T R_{j-i} k_j, \tag{11}$$

$$logits(i,j) = s(i,j) - m * T(i,j), \tag{12}$$

$$H\text{-}T_{i,j} = \begin{cases} 1, & \text{if } logits(i,j) \geq 0, \\ 0, & \text{if } logits(i,j) < 0, \end{cases} \tag{13}$$

$$Loss_{AI} = log\left(1 + \sum_{(i,j)\in P} e^{-logits(i,j)}\right) + log\left(1 + \sum_{(i,j)\in Q} e^{-logits(i,j)}\right). \tag{14}$$

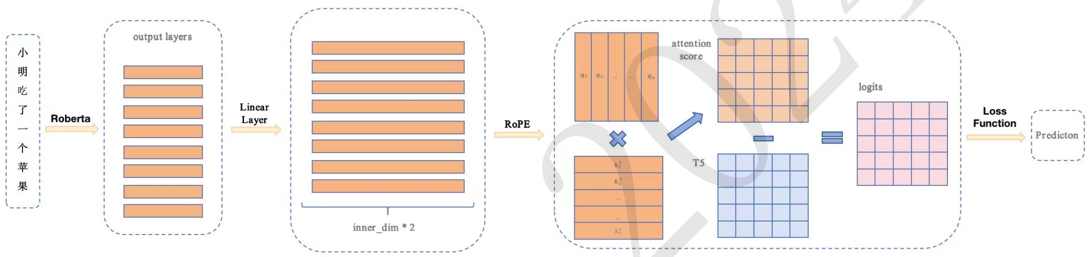

Figure 6: Argument Extraction Model

4.3 Task 3: Role Identification

For the task of the role identification, our paper fully leverages the prediction results from the frame classification module and the argument identification module. Based on the identified argument spans, it further predicts the role classification of each identified argument for a given target word. This task involves finding the corresponding argument c_t from the argument role set $C = \{c_1, c_2, \cdots, c_T\}$, which is essentially similar to $Task1$ as a classification task.

In our paper, we "wrap" the target word with $SpecialToken$ such as $< t >$ and $< /t >$. For different sentences, the number of targets and the relative position between targets and arguments are various, so we have to deal with following scenarios.

The scenarios for when there is one target word are divided into two cases as shown in Figure 7: the target word is either before or after the target argument.

For the case where there are two target words, it is divided into three scenarios as shown in Figure 8: both target words are before the target argument, both target words are after the target argument, and the target argument is between the two target words. At this point, we have implemented the process of incorporating the frame classification module's prediction results as frame features and the argument identification module's prediction results as argument features into the input text. These features are then successfully restored in the final prediction results to obtain the predicted outcome.

The overall model architecture for $Task3$ is shown in Figure 9. Similar to previous tasks, it involves a pre-trained model and a linear layer to produce results through argument frame activation and the RoPE process. However, before applying attention score regularization, the ALiBi method is used to enhance

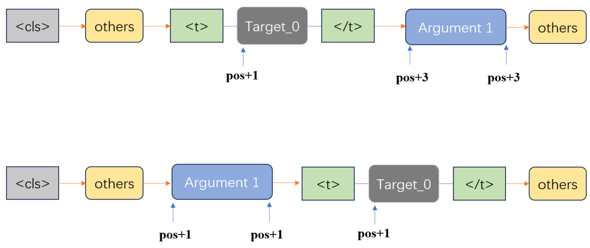

Figure 7: one target word

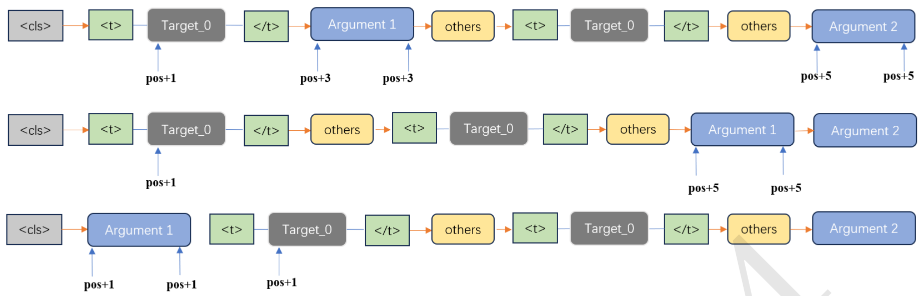

Figure 8: two target words

relative position information. The loss function for $Task3$ has the same form as the loss function in $Task1$, as shown in Equation (10).

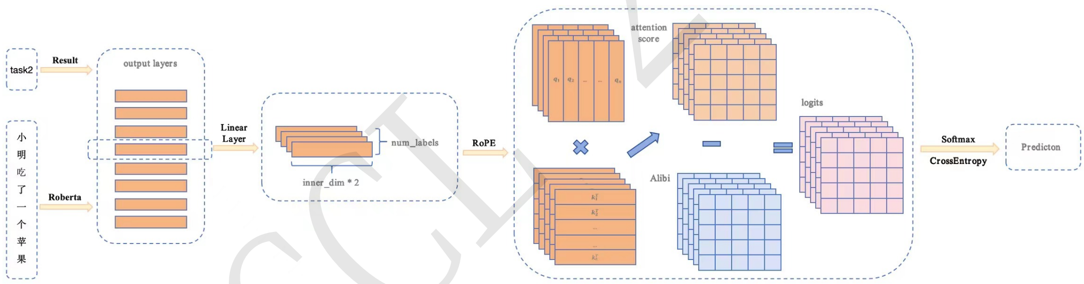

Figure 9: Role Identification Model

5 Experiments

5.1 Selection of Bert Hidden Layers

The BERT model consists of multiple layers of Transformer encoders, where the hidden layers can be divided into different levels, with preceding layers (F,indicating the initial hidden layers) and succeeding layers (L,indicating the final hidden layers) carrying different types of information. Lower-level hidden layers (F) typically capture more local and fundamental lexical-level language features, such as word context and relationships between words, and can be regarded as feature extractors to some extent. As the depth increases, hidden layers (L) gradually capture more abstract and semantically rich information, capable of extracting sentence-level language features.

In the experimental section, we compared the effects of different compositions of BERT output layers (averaging the last four layers $L1 + L2 + L3 + L4$, averaging the first two layers and the last two layers $L1 + L2 + F1 + F2$, and using only the last layer $L1$) on the model's performance on the validation set. The results are shown in Table 2.

hidden layers	task1 acc	task2 f1	task3 f1
L1	0.732	**0.791**	0.688
L1 + L2 + L3 + L4	**0.738**	0.778	**0.749**
L1 + L2 + F1 + F2	0.721	0.783	0.692

Table 2: Performance for Different Hidden Layer Configurations

It can be observed that, for classification tasks: $task1$ and $task3$, averaging the last four hidden layers as the output layer for BERT yields the best performance. However, for the extraction task: $task2$, directly selecting the last hidden layer as the output layer for BERT achieves the best performance.

5.2 Selection of Linear Warm-up Parameters

By using learning rate warm-up, the initial learning rate during training is kept small, allowing the model to gradually adapt to the training process. The learning rate is then gradually increased to reach a predefined higher value, thereby improving training efficiency and model performance. Through experimentation, we identified the optimal proportion of the linear warm-up phase (in the total training phase) that performs best on the validation set. The results are shown in Table 3.

T proportion	task1 acc	task2 f1	task3 f1
0.02	0.666	0.742	0.701
0.05	0.678	0.788	0.721
0.1	**0.738**	**0.791**	**0.749**
0.12	0.733	0.776	0.747
0.15	0.735	0.751	0.732
0.2	0.729	0.744	0.725

Table 3: Performance for Different Warm-up Percentage

According to the table above, setting the linear warm-up phase in the as first 10% of the total training phase yields the best performance for the model.

5.3 Selection of FGM Adversarial Training Hyper-parameters

The basic idea of FGM is to generate adversarial perturbations by computing gradients of the loss function with respect to the input data based on the original data, and then adding the perturbation to the original data. Typically, the magnitude of the perturbation is controlled by a hyperparameter called ϵ. We found the best ϵ on the validation set through implementation, and the experimental results are shown in Table 4.

ϵ	task1 acc	task2 f1	task3 f1
0.5	0.711	0.758	0.723
0.75	0.733	0.767	0.733
1	**0.738**	**0.791**	**0.749**
1.5	0.698	0.777	0.742
2	0.694	0.759	0.745

Table 4: Performance for Different ϵ

After experimentation, it was found that setting the perturbation magnitude to 1 resulted in the best performance for all three tasks.

5.4 Model Architecture Comparison

For the selection of which type of position encoding to use for each task, this paper conducted ablation experiments, including experiments with the $basemodel$, $basemodel + ALiBi$, $basemodel + ALiBi + Voting$, $basemodel + T5$, and $basemodel + T5 + Voting$.

First, we compared two base models. We found that BERT performed better on $task1$. In contrast, RoBERTa showed superior performance on $task2$ and $task3$. Additionally, incorporating positional encodings into the RoBERTa model further improved the results.

For $task1$, whether switching to the RoBERTa model or adding various positional encodings, the performance was not as good as the base model. However, for $task2$ and $task3$, simply switching to the RoBERTa model almost always yielded better results than adding various positional encodings to the base model. Therefore, most of our experiments were based on the RoBERTa model for various treatments.By combining RoBERTa with ALiBi, T5, ALiBi+Voting, and T5+Voting, where the first two methods add information from different relative position methods and the last two methods combine the punishment of relative position information with different weights through an ensemble approach, it was found that there was no significant improvement in performance, and even a slight decrease. By further comparing these effects in the T5 model experiments, we had the following conclusions displayed in table 5. It was found that in $task2$, $RoBERTa + T5$ performed the best, with an F1 score reaching a maximum of 0.791. In $task3$, $RoBERTa + ALiBi$ performed the best, reaching 0.749.

Model Architecture	task1 acc	task2 f1	task3 f1
BERT	**0.738**	0.781	0.735
RoBERTa	0.711	0.786	0.738
RoBERTa+ALiBi	0.722	0.785	**0.749**
RoBERTa+ALiBi+Voting	0.715	0.784	0.739
RoBERTa+T5	0.732	**0.791**	0.743
RoBERTa+T5+Voting	0.728	0.787	0.740

Table 5: Performance for Different Model

5.5 Final Model

We determined the best model architectures as well as their hyper-parameters for each sub-task through grid search,the result is shown in the Table 6.

Sub-task	Model Architecture	Hidden Layers	T proportion	ϵ	performance
Frame Identification	BERT	L1+L2+L3+L4	0.1	1	0.738
Argument Identification	RoBERTa + ALiBi	L1	0.1	1	0.791
Role Identification	RoBERTa + T5	L1+L2+L3+L4	0.1	1	0.749

Table 6: Final Model

6 Conclusion

Our article first acknowledges a drawback of traditional Transformer models regarding position encoding, namely the "lack of relative positional information." It proposes addressing this issue by incorporating directional information between tokens into entity classification models through the use of "$T5$" and "$ALiBi$" position encodings. Additionally, it considers scenarios involving multiple target words and enhances the performance of the model in the context of framework identification tasks through aggregation. Through experimentation, comparisons were made regarding various hyperparameters such as the composition of BERT's hidden layers, linear warm-up parameters, adversarial training parameters, and various model architectures. The best combination on the validation set was identified, resulting in achieving the fourth position on the closed-track B leaderboard. Our article also has some shortcomings. For instance, in the framework identification task, we were unable to find downstream model architectures that could effectively leverage RoBERTa. As a result, we had to settle for using BERT only, which led to a performance gap compared to the first-place result. Improving the framework identification model to obtain more accurate and comprehensive framework semantic parsing results is a direction worthy of further research.

References

Ofir Press, Noah A. Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Adam Roberts, Colin Raffel, Katherine Lee, et al. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. Google, Tech. Rep.

Jie Su, Mairin Ahmed, Yutong Lu, et al. 2024. Roformer: Enhanced Transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Su X, Li R, Li X, et al. 2021. A knowledge-guided framework for frame identification. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5230-5240.

Zhou S, Xia Q, Li Z, et al. 2021. Fast and accurate end-to-end span-based semantic role labeling as word-based graph parsing. arXiv preprint arXiv:2112.02970.

Li R, Zhao Y, Wang Z, et al. 2024. A Comprehensive Overview of CFN From a Commonsense Perspective. *Machine Intelligence Research*, pp. 1-18.

Liu Y, Li Z, Zhang M. 2023. CCL23-Eval 任务 3 系统报告: 苏州大学 CFSP 系统 (System Report for CCL23-Eval Task3: SUDA CFSP System). In: *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 3: Evaluations)*, pp. 84-93.

Li Z, Guo X, Qiao D, et al. 2023. CCL23-Eval 任务 3 系统报告: 基于旋转式位置编码的实体分类在汉语框架语义解析中的应用 (System Report for CCL23-Eval Task 3: Application of Entity Classification Model Based on Rotary Position Embedding in Chinese Frame Semantic Parsing). In: *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 3: Evaluations)*, pp. 94-104.

Huang S, Shao Q, Li W. 2023. CCL23-Eval 任务 3 系统报告: 基于多任务 pipeline 策略的汉语框架语义解析 (System Report for CCL23-Eval Task 3: Chinese Frame Semantic Parsing Based on Multi-task Pipeline Strategy). In: *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 3: Evaluations)*, pp. 105-112.

苏剑林. 2021. 让研究人员绞尽脑汁的Transformer位置编码.

郝晓燕, 刘伟, 李茹, 刘开瑛. 2007. 汉语框架语义知识库及软件描述体系. 中文信息学报, 05: 96-100+138.

李济洪. 2010. 汉语框架语义角色的自动标注技术研究. 博士学位论文. 山西大学.

李济洪, 高亚慧, 王瑞波, 李国臣. 2011. 汉语框架自动识别中的歧义消解. 中文信息学报, 25(03):38-44. 山西大学计算中心, 山西大学数学科学学院, 太原工业学院.

石佼, 李茹, 王智强. 2014. 汉语核心框架语义分析. 中文信息学报, 28(06):48-55.

赵红燕, 李茹, 张晟, 等. 2016. 基于DNN的汉语框架识别研究. 中文信息学报, 30(06):75-83.

王晓晖, 李茹, 王智强, 等. 2022. 基于 Self-Attention 的句法感知汉语框架语义角色标注. 中文信息学报, 36(10):38-44.