A GlobalPointer based Robust Approach for Information Extraction from Dialog Transcripts

Yanbo J. Wang[1], **Sheng Chen**[1], **Hengxing Cai**[2], **Wei Wei**[3], **Kuo Yan**[1], **Zhe Sun**[1], **Hui Qin**[1], **Yuming Li**[4] and **Xiaochen Cai**[5]
[1]LYZD-FinTech Co., LTD, Beijing, China
[2]4Paradigm Inc., Beijing, China
[3]School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China
[4]The University of Auckland, Auckland, New Zealand
[5]Nanjing University, Nanjing, China

Abstract

With the widespread popularisation of intelligent technology, task-based dialogue systems (TOD) are increasingly being applied to a wide variety of practical scenarios. As the key tasks in dialogue systems, named entity recognition and slot filling play a crucial role in the completeness and accuracy of information extraction. This paper is an evaluation paper for SereTOD 2022 Workshop challenge (Track 1: Information extraction from dialog transcripts). We proposed a multi-model fusion approach based on GlobalPointer, combined with some optimisation tricks, finally achieved an entity F1 of 60.73, an entity-slot-value triple F1 of 56, and an average F1 of 58.37, and got the highest score in SereTOD 2022 Workshop challenge[1].

1 Introduction

Task-oriented dialogue (TOD) systems are designed for specific application areas and have gained more and more attention in both academia and industry recently (Gao et al., 2019).

As a branch of the dialogue systems, TOD systems are different from question-and-answer (QA) systems and chat-oriented dialogue systems. TOD system needs to determine the user's intent through understanding, analysis, information extraction, and clarification. Then complete a round of dialogue through natural language generation or APIs.

According to the work of Zhao et al. (Zhao and Eskenazi, 2016) and Zhang et al. (Zhang et al., 2020), the structure of a traditional TOD system is shown in Figure 1, which can be divided into three modules, Spoken Language Understanding (SLU), Natural Language Generation (NLG), and Dialogue Manager.

The SLU Module converts language into semantic representations, the purpose is to obtain the semantic information of user input speech. The

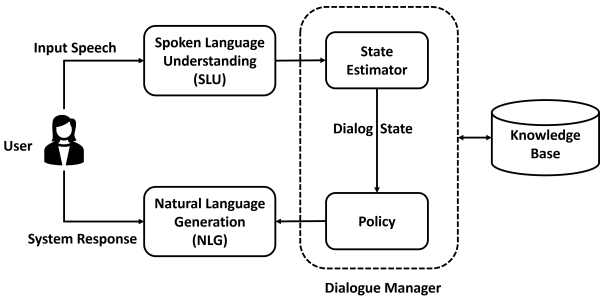

Figure 1: The data samples in the product catalogue in the Shopping Queries Data Set.

downstream module of SLU is the dialogue manager module. The task of this module is to decide how the system responds to the input speech (McTear, 2004) and then the system updates its internal state, and then the system determines the system behaviour through policies. In order to provide information to the user, the dialogue manager usually needs to query the knowledge base or the Internet, and it also needs to consider the historical data in the multi-round dialogue. Finally, the NLG module translates the decisions of the system into natural language-based dialogues. Among them, the state variables contain variables that track the dialogue process, as well as slots that represent user needs.

1.1 Task description

The task of SereTOD 2022 Workshop challenge consists of 2 tracks, and we focus on track 1 (Information extraction from dialogue transcripts) in this paper. There are four sub-tasks for track 1:

- **Entity Extraction.** Extract entity mentions in real-life dialogues according to the entity types defined in the schema (including related data package plan and services, a total of nine categories).
- **Entity Coreference Resolution.** Since an entity might be mentioned in different surface

[1]https://docs.google.com/spreadsheets/d/1w28AKkG6Wjmuo15QlRlRyrnv859MT1ry0CHV8tFxY9o/edit?usp=sharing

Proceedings of the Towards Semi-Supervised and Reinforced Task-Oriented Dialog Systems (SereTOD), pages 13 - 18

```
{
  "id": "94bba9d63c097df1800482d827287e47",
  "content": [
    {
      "[SPEAKER 1]": "你好，很高兴为您服务。",
      "[SPEAKER 2]": "我你好，我想问一下，我想办那个半年_六个G那个包，我想问问那个包那个不是属于全国漫游嘛",
      "客服意图": "问候",
      "用户意图": "问候,求助-查询（ent-1-流量范围）",
      "info": {
        "ents": [
          {
            "name": "半年_六个G那个包",
            "id": "ent-1",
            "type": "流量包",
            "pos": [
              [
                2,
                15,
                24
              ]
            ]
          }
        ],
        "triples": [
          {
            "ent-id": "ent-1",
            "ent-name": "半年_六个G那个包",
            "prop": "业务时长",
            "value": "半年",
            "pos": [
              2,
              15,
              17
            ]
          },
          {
            "ent-id": "ent-1",
            "ent-name": "半年_六个G那个包",
            "prop": "流量总量",
            "value": "_六个G",
            "pos": [
              2,
              17,
              21
            ]
          }
        ]
      }
    }
  ],
```


Figure 2: A basic unit of the MobileCS (mobile customer-service) dialog dataset.

forms, for example, "100元的流量包", "那个流量包", "100元的那个业务", "刚才那业务" may refer to the same entity " 100元流量包 (100 Yuan data package plan)". Thus we need to represent the entities with coreference relationships in a unified id.

- **Slot Filling.** Extract the slot value corresponding to the entity slots (including the specific content of the package or business and the status of the user, etc.). For example, in the dialogue "10GB套餐的月费用是50元 (The price for the 10GB data package plan is 50 Chinese Yuan per month)", "50元 (50 Chinese Yuan)" will be the value for the monthly price slot.

- **Entity Slot Alignment.** Align entities and slot values with corresponding relationships.

1.2 Data description

The data for this challenge is ***MobileCS (mobile customer-service) dialog dataset*** (Ou et al., 2022) around 100K dialogues (in Chinese), which come from real-world dialogue transcripts between real users and customer-service staffs from China Mobile, with privacy information anonymised.

The official data includes three parts: training data, dev data and test data. A basic unit of the data sample is shown in Figure 2. In which Speaker ID such as "**[SPEAKER 1]**" and "**[SPEAKER 2]**" refer to the speaker of the dialogue, "用户意图" represents the user intent, "客服意图" represents the system intent, the entities and triples are the information mentioned in this turn.

2 Approach

In this paper, we focus on the baseline (Liu et al., 2022) and practical business difficulties of and dialogue system, and propose suitable solutions. The difficulties can be summarised as follows:

- In the slot value extraction stage, the length of the slot value to be extracted is relatively long, the categories are complex, and the general sample repetition is relatively small. Especially for the categories '业务规则' and '持有套餐'.

- The problem of label scope coverage nesting. Labels from class A may be overwritten by labels from class B.

- The distribution of training data, dev data, and test data has an obvious difference.

- Some single-turn dialogues with entities contain very little information, but there are many entities containing business rules need to be identified.

According to the above-mentioned difficulties, our solutions can be summarised as follows:

- We apply the GlobalPointer to the Entity Extraction and Slot Filling tasks , set different loss weights for positive and negative samples.

- Data pre-processing： The addition of global context information, split the paragraphs into single characters, merge the original training data and dev data to train.

- We add training data and dev data to the Pretrained Masked Language Model.

- We optimized the Entity Slot Alignment task to increase the cross-validation score by 9 percentage points.

- In the Entity Extraction task, we trained some models with different maximum token length (384, 256, 280). The differences between models bring benefits to fusion.

- We truncate the 256 × 256 token probability matrix according to the maximum entity length and fuse it to greatly reduce memory consumption.
- For the overlapping nested entities in the Entity Extraction task, we do post-processing to eliminate them.

2.1 Model and tricks

In the challenge, we found that nested entities and non-nested entities coexist in training data and dev data. The sequence-to-sequence method in baseline cannot handle the situation of nested entities, therefore, we use the end-to-end method to solve this tough issue. In entity extraction and slot filling, we are mainly based on GlobalPointer (Su et al., 2022), a novel efficient span-based approach for named entity recognition, which uses global normalisation for named entity recognition, and can identify nested and non-nested entities indiscriminately.

For any sentence, GlobalPointer constructs an upper triangular matrix to traverse all valid spans, as shown in Figure 3, each grid corresponds to an entity span. Assuming that after the input sentence

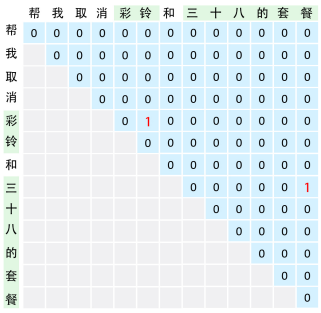

Figure 3: Schematic diagram of GlobalPoniter multi-head identification of nested entities.

passes through the encoder, the representations at positions i and j are h_i and h_j, and the query vector q_i and key vector k_j of the two are obtained through the fully connected layer:

$$q_i = W_q h_i + b_q$$

$$k_j = W_k h_j + b_k$$

Then the score of each span $s(i, j)$ predicted as an entity is:

$$s(i, j) = q_i^T k_j$$

On this basis, GlobalPointer incorporates the Rotational Position Encoding (RoPE) mechanism to explicitly introduce relative position information to the prediction of span pairs. For position m, RoPE calculates an orthogonal matrix R_m, then multiply R_m by q to rotate q. According to the matrix multiplication rule, if k is also multiplied by the RoPE. At this time, the score $s(i, j)$ of the span will have relative position information R_{n-m}:

$$(R_m q_i)^T (R_n k_j) = q_i^T R_m^T R_n k_j = q_i^T R_{n-m} k_j$$

2.1.1 Loss function

Since the number of entities in the sentences in the dataset is very small and there are a large number of negative samples, we do not use binary classification in our method but designed a multi-label loss function. For identifying entities of a specific class α, the fragments with $s_\alpha(i, j) > 0$ are regarded as the output of entities of type α. The loss function is:

$$log(1+\sum_{(i,j)\in P_\alpha} e^{-s_\alpha}(i,j)) + log(1+\sum_{(i,j)\in Q_\alpha} e^{s_\alpha}(i,j))$$

Where P_α is a set of spans with entity type α in the dataset, Q_α is a set of spans that are not entities or whose entity type is not α in the sample, we only need to consider the combination of $i \leq j$, which is the upper triangular matrix in the blue area in Figure 2.

$$\omega = \{(i, j) | 1 \leq i \leq j \leq n\}$$

$$P_\alpha = \left\{(i, j) | t_{[i:j]} \in \alpha\right\}$$

$$Q_\alpha = \Omega - P_\alpha$$

Due to the low accuracy in the entity extraction stage, we increase the loss weight of positive samples and decrease the loss weight of negative samples, which can increase F1 by about one percentage point. However, slot filling cannot effectively improve the model accuracy through different loss weights.

2.1.2 How to use the dev data

In view of the large difference in the distribution of training data, dev data, and test data, how to use dev data is also a key factor to ensure that the model can perform well in test data. First, we locate the position of the slot value of the official dev data. However, some position tags are difficult to capture, so we eliminate them in the training phase. Then we will merge and disarrange dev data and training data, and divide them into four folds. Finally, we will apply the split data to each stage of the pipeline.

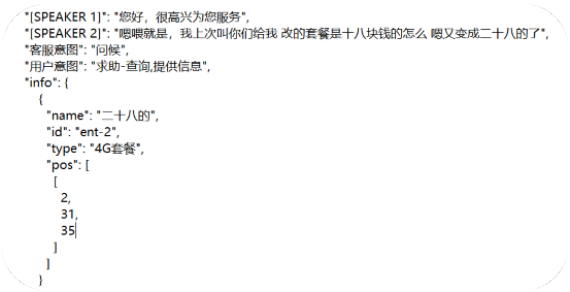

Figure 4: The data sample from challenge data.

2.1.3 The addition of global context information

In the challenge data, the information obtained by simply concatenating **[SPEAKER1]**, and **[SPEAKER2]** cannot accurately identify the entity type. As shown in Figure 4, only by adding context information, we can make it clear that the name value "二十八的" refers to "套餐" or "4G套餐". During the optimisation process, we show that adding global context information can improve a single model by about 2 percentage points.

Since we take the current dialogue content and the global context concatenating as input, we mask the concatenated tokens through attention_mask. However, the global context information is only used as enhancement content and does not participate in the calculation of loss, so we mask the context information to calculate the effective loss, as shown in Figure 5.

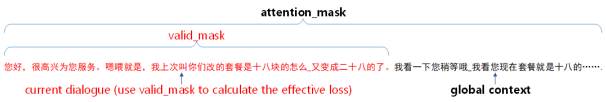

Figure 5: Attention_mask based on the global context and valid_mask for the current conversation.

2.1.4 Break down context by character

There are many phonetic expressions related to place names and special terms in the training and dev data of the challenge data. As shown in Figure 6, according to the Chinese BERT (Devlin et al., 2018) tokenizer, the Chinese pinyin may be split into words that have nothing to do with semantics, so we first split the paragraphs into single characters and then send them to the BERT tokenizer.

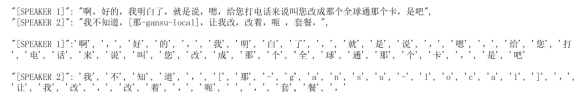

Figure 6: The sample data of place names with pinyin expressions and result of segmentation.

2.1.5 Pre-trained Masked Language Model (MLM)

We add training data and dev data to the Pre-trained Masked Language Model, and use the pre-trained model for entity extraction and slot filling, which increases by about 1 percentage point.

2.1.6 Entity slot alignment task optimisation

In the Baseline (Liu et al., 2022) given by the challenge, when calculating the similarity between any entity (ent) and any slot value (triple), there is some noise affecting the model training, For example, when calculating the similarity between **ent1** and **triple1**, such as **<entity>ent1<entity>... <entity>ent2<entity>... <slot>triple1<slot>**. In this case, other types of entities appear in the text between **ent1** and **triple1** will also be labelled, which will cause interference in training and dev. Therefore, in the face of this situation, we remove the entity tag **<entity>** related to **ent2**, which can directly improve the model verification result by 9 percentage points.

2.1.7 Post-processing

In the post-processing stage, there are many overlapping entities in our entity extraction part. In this case, we choose the one with the highest probability as the optimal choice. For example, in the following cases shown in Figure 7, we will remove ent1 and select ent2:

ent1:{"pos": [[1,20,25]],"type": "附加套餐","name": "十块钱套餐",prob:0.6}
ent2:{"pos": [[1,20,25]],"type": "套餐","name": "十块钱套餐",prob:0.9}

Figure 7: An example case for post-processing.

3 Model fusion

In terms of data selection and split-folding strategy, we merge the original training data and dev data, and split them into four folds for training through k-fold. In Table 1 and Table 2, the score is calculated on the out-of-fold of the combination of training and dev data.

3.1 Entity extraction

In the entity extraction subtask, we selected five models of Roformer (Su et al., 2021), DeBERTa (He et al., 2020), RoBERTa (Liu et al., 2019), MacBERT (Cui et al., 2020), and NEZHA (Wei et al., 2019) for probability average fusion, and found that the fusion of models with different token lengths can achieve better results. We chose models with a maximum token length of 256, 280, and 384 for fusion; at the same time, we also chose to add Efficient GlobalPointer (Su et al., 2022) to the fusion to increase the difference. The final fusion result (mean average of probability) is 1.3 percentage points higher than the highest single model. The result is shown in Table 1.

Backbone	Head	Max Length	4 fold F1	Ensemble F1
roformer	Efficient_GlobalPointer	384	0.557	0.570
deberta	GlobalPointer	280	0.556	
nezha	GlobalPointer	256	0.547	
roberta	GlobalPointer	256	0.547	
macbert	GlobalPointer	256	0.549	

Table 1: The model fusion result of entity extraction.

3.2 Slot filling

In the slot filling subtask, we selected four models of Roformer (Su et al., 2021), RoBERTa (Liu et al., 2019), MacBERT (Cui et al., 2020), and NEZHA (Wei et al., 2019) for probability average fusion. The final fusion result is 0.9 percentage points higher than the highest single model. The result is shown in Table 2.

Backbone	Head	Max Length	4 fold F1	Ensemble F1
roformer	GlobalPointer	256	0.607	0.616
nezha	GlobalPointer	256	0.605	
roberta	GlobalPointer	256	0.600	
macbert	GlobalPointer	256	0.602	

Table 2: The model fusion result of slot filling.

3.3 Entity coreference resolution and entity slot alignment

Due to the time limit of the challenge, the 4-fold and 5-fold models were not trained for these two tasks. First, we cut the original data into 4 folds, and merge three fold data and dev data as training data to obtain model 1. Then we cut the original data into 5 folds, and merge four fold data and dev data as training data to obtain model 2. The final submission is a probability average fusion of model1 and model2. The scores are in Table3.

Entity Coreference Resolution	
4-fold	***5-fold***
0.887	0.891
Entity Slot Alignment	
4-fold	***5-fold***
0.884	0.891

Table 3: The 4-fold and 5-fold result for resolution and alignment.

3.4 GlobalPointer fusion matrix optimisation

In the GlobalPointer fusion stage, a four-dimensional ($sample\_num \times type\_num \times L \times L$) matrix is generated. The last two dimensions are the maximum token length of 256. Since the final matrix is too large and there are many models, the memory cost of Numpy storage and calculation is too high, especially in the slot filling stage. Therefore, we first initialise a Numpy matrix ($sample\_num \times type\_num \times L \times max\_lengh\_entity$), in which max_lengh_entity is the maximum entity length, which is 20 or 50, which is much smaller than 256. This dimension data is obtained by truncating the matrix. And through the calculation of the model, the probability matrix is continuously filled, reducing the number of variables in the memory, and finally obtaining the final result. One example is shown in Figure 8.

4 Conclusion

In this challenge, we use the GlobalPointer-based structure and probabilistic average fusion of Roformer, DeBERTa, RoBERTa, MacBERT, and NEZHA as the main solution. At the same time, we adopted tricks such as adding global context information and breaking down context by character in the following step to further optimise the results. Finally, we end up with an entity F1 of 60.73, an entity-slot-value triple F1 of 56, and an average F1 of 58.37, and got the highest average F1 score in the challenge of SereTOD 2022 Workshop.

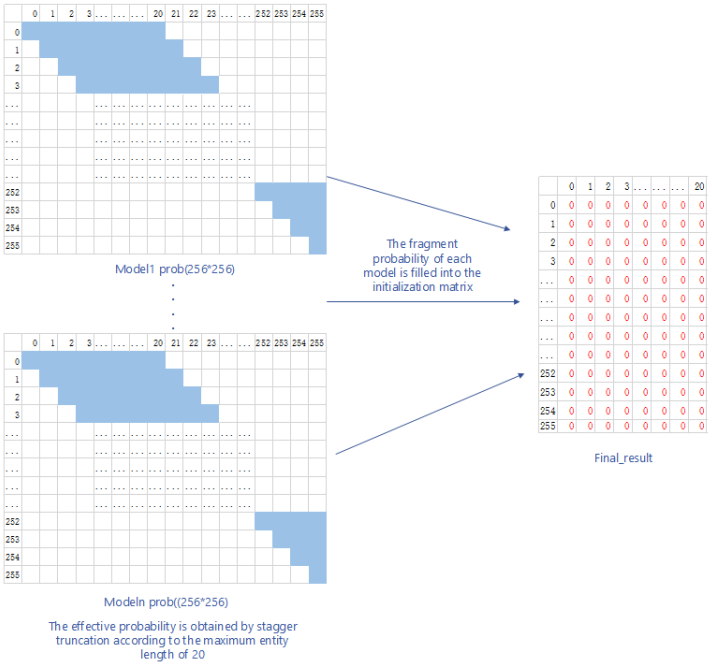

Figure 8: EntityExtraction task probability fusion – get effective probability by stagger truncation with maximum entity length of 20.

References

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. *arXiv preprint arXiv:2004.13922*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jianfeng Gao, Michel Galley, and Lihong Li. 2019. *Neural approaches to conversational AI: Question answering, task-oriented dialogues and social chatbots*. Now Foundations and Trends.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Hong Liu, Hao Peng, Zhijian Ou, Juanzi Li, Yi Huang, and Junlan Feng. 2022. Information extraction and human-robot dialogue towards real-life tasks: A baseline study with the mobilecs dataset. *arXiv preprint arXiv:2209.13464*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Michael F McTear. 2004. *Spoken dialogue technology: toward the conversational user interface*. Springer Science & Business Media.

Zhijian Ou, Junlan Feng, Juanzi Li, Yakun Li, Hong Liu, Hao Peng, Yi Huang, and Jiangjiang Zhao. 2022. A challenge on semi-supervised and reinforced task-oriented dialog systems. *arXiv preprint arXiv:2207.02657*.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Jianlin Su, Ahmed Murtadha, Shengfeng Pan, Jing Hou, Jun Sun, Wanwei Huang, Bo Wen, and Yunfeng Liu. 2022. Global pointer: Novel efficient span-based approach for named entity recognition. *arXiv preprint arXiv:2208.03054*.

Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. 2019. Nezha: Neural contextualized representation for chinese language understanding. *arXiv preprint arXiv:1909.00204*.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027.

Tiancheng Zhao and Maxine Eskenazi. 2016. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. *arXiv preprint arXiv:1606.02560*.