PALI at SemEval-2022 Task 7: Identifying Plausible Clarifications of Implicit and Underspecified Phrases in Instructional Texts

Mengyuan Zhou and **Dou Hu** and **Mengfei Yuan** and **Meizhi Jin**
and **Xiyang Du** and **Lianxin Jiang** and **Yang Mo** and **Xiaofeng Shi**
Ping An Life Insurance Company of China, Ltd.
{ZHOUMENGYUAN425, HUDOU470, YUANMENGFEI854, JINMEIZHI005,
DUXIYANG037, JIANGLIANXIN769, MOYANG853, SHIXIAOFENG309}
@pingan.com.cn

Abstract

This paper describes our system used in the SemEval-2022 Task 7(Roth et al.): Identifying Plausible Clarifications of Implicit and Underspecified Phrases. Semeval Task7 is an more complex cloze task, different than normal cloze task, only requiring NLP system could find the best fillers for sentence. In Semeval Task7, NLP system not only need to choose the best fillers for each input instance, but also evaluate the quality of all possible fillers and give them a relative score according to context semantic information. We propose an ensemble of different state-of-the-art transformer-based language models(i.e., RoBERTa and Deberta) with some plug-and-play tricks, such as Grouped Layer-wise Learning Rate Decay (GLLRD) strategy, contrastive learning loss, different pooling head and an external input data preprecess block before the information came into pretrained language models, which improve performance significantly. The main contributions of our system are 1) revealing the performance discrepancy of different transformer-based pretraining models on the downstream task; 2) presenting an efficient learning-rate and parameter attenuation strategy when fintuning pretrained language models; 3) adding different constrative learning loss to improve model performance; 4) showing the useful of the different pooling head structure. Our system achieves a test accuracy of 0.654 on subtask1(ranking 4th on the leaderboard) and a test Spearman's rank correlation coefficient of 0.785 on subtask2(ranking 2nd on the leaderboard).

1 Introduction

Cloze tasks have become a standard framework for evaluating various discourse-level phenomena in NLP. Some prominent examples include the narrative cloze test(Hoshino and Nakagawa, 2007), the story cloze test (Xie et al., 2020), and the LAMBADA word prediction task(Paperno et al., 2016). In these tasks, NLP systems are required to make a prediction about the filler of a cloze that is most likely to continue the discourse. However, these existing cloze tasks focus on the accuray of choosen fillers, ignore evaluating the absolute quality of all possible predictions.

The goal of Semeval 2022 Task7 is to evaluate the ability of NLP systems to distinguish between plausible and implausible clarifications of an instruction. The task is formulated as a complex cloze task, which involve two sub tasks. In Sub task 1, for each(sentence, filler) pair, NLP system need to classify four fillers into plausible, implausible or neutral and the evaluating indicator is accuracy. In Sub task 2, for each pair, NLP system need to predict scores for five fillers and the evaluating indicator is Spearman's rank correlation coefficient.

Since 2018, NLP models have adopted the concept of pre-training on a diverse corpus of unlabelled text, followed by supervised finetuning on specific tasks. Pretrained models are built to simulate anthropomorphic learning, wherein existing knowledge can be adapted to new tasks without doing any training on these tasks from scratch - a requirement of traditional machine learning models. By now, these pre-trained large language models such as Bert(Devlin et al., 2018), Roberta(Liu et al., 2019),Xlm-roberta (Conneau et al., 2019), Deberta(He et al., 2021) has been widely used to solve all kinds of language understanding tasks. Additionally, fine-tuning self-supervised pre-trained models has significantly boosted state-of-the-art performance on natural language processing (NLP) tasks. Many evidence showing models with pre-trained commonsense knowledge can be well applied in the field of cloze task(Cui et al., 2020), because cloze task needs commonsense language knowledge and genera language knowledge.

Additionally, there are many training tricks can be used to improve the performance and generalization ability of the large pre-trained language models. First, adding contrastive learning loss in

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1084 - 1089

supervised task, such as Ntxent loss(Chen et al., 2020). Second, in case of pretrained language models' catastrophic forgetting in funtuning period, we set different learning rate and weight decay rate for different pre-trained language model layers(Zhang et al., 2021). Third in order to get the best sentence embedding, trying different pooling head is necessary. Inspired by these discoveries, we designed two NLP system for sub-task1 and sub-task2.

2 Task Setup

Formally, each instance in dataset is composed of 5 sentences, 5 fillers, 1 clarification phenomena and 1 score:

- $Article\ title$: title of the wikiHow article in which the sentence occurs.
- $Section\ header$: heading of the section which the sentence is part of.
- $Previous\ context$: a couple of sentences that occur before the sentence in question - omissions are marked by "(...)".
- $Sentence$: the sentence with a placeholder "..." that marks where the fillers should be inserted.
- $Follow-up\ context$: a couple of sentences that occur after the sentence in question - omissions are marked by "(...)".
- $Filler1 - Filler5$: the five different fillers.
- $Score$: is the quality score of each candidate word, range from 1 to 5.
- $Resolved\ pattern$: name of the clarification phenomenon (cf. list above: implicit reference, fused head, added noun, metonymic reference.)

As showing in figure 1, this task is a cloze task, using fillers(Filler1 to Filler5) to insert a blank at the position in the text(e.g. Screw each stringer to ___ the deck frame with a drill, use L-brackets and deck screws to attach the stringers to the deck.). Additional, article title(), section header,previous and fellow-up sentence, resolved pattern are given.

Sub-task 1 is a classification sub-task, the evaluation metrics is overall accuracy. According to rules, converting each real-valued gold score to a class label as follows:

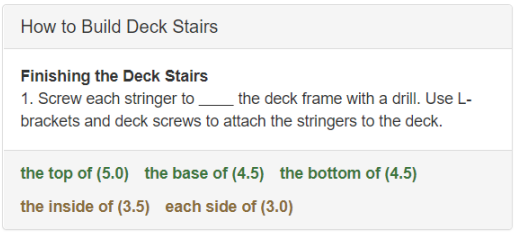

Figure 1: Data Instance.

- $score <= 2.5$: IMPLAUSIBLE
- $2.5 < score < 4$: NEUTRAL
- $score >= 4$: PLAUSIBLE

Subtask2 is a regression subtask, our system need to predict each instance's gold plausibility score. The submissions will be scored based on Spearman's rank correlation coefficient which compares the predicted plausibility ranking over all test instances with the gold ranking.

3 Data Summary and Analysis

3.1 Data Construction

We try two different methods to preprocess input data.

Strategy One: We simplely concatenate resolved pattern, article title, section header, previous context, sentence and follow-up context first. And then fill the blank spaces with 5 fillers separately. Finally in order to highlight fillers information, we add special symbols "<e>" before and after fillers.

Strategy Two: Since Resolved pattern is kind of category feature which is different than the other text features, We first replace resolved pattern with their explanations. And then connecting explanations to the other information. Finally, adding special symbols "<e>" before and after fillers. The explanation of resolved pattern showing below:

MPLICIT REFERENCE: In the original version of a sentence, there is an implicit reference to a previously mentioned entity. The revision makes this reference explicit.

FUSED HEAD: In the original version, there is a noun phrase where the head noun is missing. The revision adds that noun.

ADDED NOUN: The revision adds a compound modifier to a noun to make its meaning more specific.

METONYMIC REFERENCE: In the original version, a noun is used in a metonymy. The revision

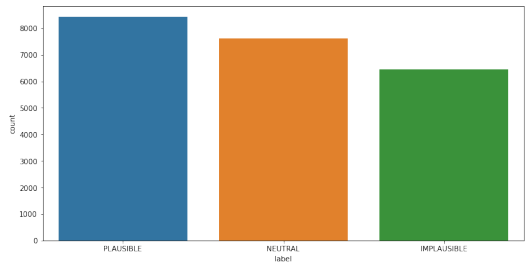

Figure 2: Label Distribution.

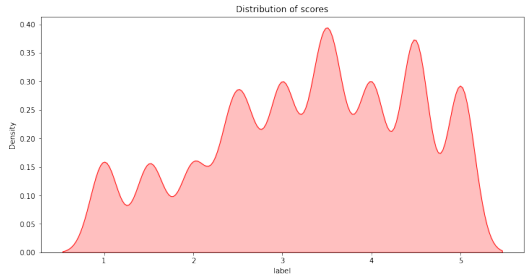

Figure 3: Score Distribution.

makes the particular component or aspect of a noun explicit that is meant.

We adopt two kinds of preprocess methods in our experiment and Strategy Two showing a better performance.

3.2 Data Analysis

We can see the label distribution and score distribution in Figure 2 and Figure 3. The largest number of labels is a plausible and the average score is near 3.5.

4 Methodology

For this task, we have tried a variety of modeling, optimization methods, learning rate schedule and different constrative learning loss. Details are described below.

4.1 Model Design

We design same model architectures for this two sub-task. Our model is based on different pretrained models, such as roberta, xlmroberta and deberta. After these pre-trained block, we set different pooling head to replace cls head in order to get better sentence information. We try 3 different head, mean-max pooling head, cls head and lstm + attention head, and all the pooling structures showing good performance in our experiments.

CLS Pooling Head: CLS Pooling Head is the last layer hidden-state of the first token of the sequence (classification token) further processed by a Linear layer and a Tanh activation function. The Linear layer weights are trained from the next sentence prediction (classification) objective during pretraining. We reset cls top linear layer weights in finetuning. We believe this weights are over fitting NSP task that have a bad effect on further finetuning.

Mean-Max Pooling Head: We consider the last hidden state [batch, maxlen, hidden_state], then take max across maxlen dimensions to get max pooling embeddings. For mean pooling, we also consider the last hidden state, the average across max length dimensions to get averaged/mean embedding. Finally we concatenate this two embedding and further processed by a Linear layer.

LSTM plus Attention Pooling Head: Since LSTM network is inherently suitable for processing sequential information, we can use a LSTM network to connect all token of last hidden state [batch, maxlen, hidden_state], and the output of the all LSTM cell [batch, nums_LSTM_cell, LSTM_hidden_state] is used as input of next dot-product attention module. After dot-product attention module, we pass the pooled output to a fully connected layer for label prediction.

4.2 Training Details

Our system adopt grouped layer-wise learning rate decay(GLLRD)(Ginsburg et al., 2018) as main learning rate and weight decay strategy. GLLRD is a method that applies higher learning rates for top layers and lower learning rates for bottom layers. This is accomplished by setting the learning rate of the top layer and using a multiplicative decay rate to decrease the learning rate layer-by-layer from top to bottom.

In our experiment, We set 3 parameter group for 24-layers pretrained large language models, first group include 0 to 7 pretrained layers; second group include 8 to 15 pretrained layers; third group include 16-23 pretrained layers. We design a base learning rate 1e-5 for group 2(8-15 pretrained layers); A lower learning rate 1e-5/1.6 for group 1; A higher learning rate 1e-5 * 1.6 for group3; And a much higher learning rate(2e-4) for top layers.

The goal is to modify the lower layers that encode more general information less than the top layers that are more specific to the pre-training task. This method is adopted in fine-tuning several recent pre-trained models, including Roberta, Xlm-roberta and Deberta-v3(Zhang et al., 2021). Addi-

tionally, we adopt cosine_warmup in our learning rate scheduler and we adopt AdamW with opening bias correction. For task7 dataset, if not open bias correction, will lead to huge fluctuations in model performance.

4.3 Loss Function Design

Our system involve 3 kinds of loss, Classification loss(CrossEntropy loss), regression loss(MSE loss) and contrastive loss(NTXent loss). Inspired by recent contrastive learning algorithms, our system adopt NTXent loss as sencond loss which is proposed in SimCLR. Contrastive loss learns representations to maximize agreement between differently augmented views of the same data example in the latent space.In this task, our system need to evaluate the quality of all possible fillers, NTXent loss can help system get more robust sentence representation to classify all fillers.

For NTXent loss, We randomly sample a minibatch of N examples and define the contrastive prediction task on pairs of augmented examples derived from the minibatch, resulting in 2N data points.We do not sample negative examples explicitly. Instead, given a positive pair, similar to (Chen et al., 2017), we treat the other 2(N − 1) augmented examples within a minibatch as negative examples. Let $sim(u, v) = \mathbf{u}^\top v/||u||||v||$ denote the dot product between ℓ_2 normalized u and v (i.e.cosine similarity). Then the loss function for a positive pair of examples (i, j) is defined as (Chen et al., 2020):

$$\ell_{i,j} = -\log \frac{\exp\left(\text{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k\neq i]} \exp\left(\text{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_k\right)/\tau\right)} \quad (1)$$

where $\mathbb{1}_{[k\neq i]} \in (0, 1)$ is an indicator function evaluating to 1 iff $[k \neq i]$ and τ denotes a temperature parameter. The final loss is computed across all positive pairs, both (i, j) and (j, i), in a minibatch.

In subtask 1, we adopt the weighted average method to obtain the final loss between CrossEntropy loss and NTXent loss. The method is as follows:

$$Loss_{total} = Loss_{CE} + \alpha Loss_{NTX}, \quad (2)$$

where $Loss_{CE}$ is CrossEntropy loss,$Loss_{NTX}$ is NTXent loss, $\alpha = 0.1$

In subtask 2, we adopt the final weighted average method to obtain the final loss between MSE loss and NTXent loss. The method is as follows:

$$Loss_{total} = Loss_{MSE} + \alpha Loss_{NTX} \quad (3)$$

where $Loss_{MSE}$ is MSE loss,$Loss_{NTX}$ is NTXent loss, $\alpha = 0.1$.

5 Experiments

5.1 Experiment Settings

In order to get better performance in this few-sample dataset, We apply AdamW as an optimization algorithm with 10% steps of warmup and open the the correct_bias item (Zhang et al., 2021). For hyperparamete, we fine-tune the uncased, 24-layer $Roberta_{Large}$, $Xlm - Roberta_{Large}$ and $Deberta_{Large}$ model with batch size 40, dropout 0.2, cosine_warmup 1e-2. Additionally we adopt grouped layer-wise learning rate decay strategy with base learning rate 1e-5, weight-ratio 1.6 and a much higher learning 2e-4 for top pooling layers, mentioned in 4.2 Training Details. We used stratified k-fold method to split training data into 5 folds.

5.2 Experimental Results

We separate trained our system for sub-task1 and sub-task2. In both sub-task, we adopt sentence embedding to settle the further classification and regression works. As showing in Table 1, for each method, the score we report here is the average score of the experiment results. From Table 1, we see that the deberta-v3 model showing the best overall performance on both sub-task1 and sub-task2. In sub-task1, we can find deberta model is at least 2% higher than roberta model and 1.8% higher than xlm-roberta model. On sub-task2, deberta model's improvement is much higher, compared with roberta and xlm-robera, deberta has an improvement of more than 4.5% and 3.1% respectively. More important, data construction method 2 replacing resolved pattern with their explanations also provided a performance boost, around 0.7% in both sub-task. GLLRD strategy and contrastive loss bring a great improvement, neary 1% and 0.5%. Different pooling head also bring different influence in final score, Lstm + Attention head showing the best performance, which can reach 0.649 in sub-task1 and 0.782 in sub-task2 . Totally, after trying different method and model fusion, our system

Pretrained model	Data Construction Method	Pooling head	GLLRD	Contrastive loss	ACC @ subtask1	Spearman coefficient @ subtask2
roberta-large	Strategy One	CLS Pooling	False	—	0.602	0.696
	Strategy Two	CLS Pooling	False	—	0.608	0.704
	Strategy Two	CLS Pooling	True	—	0.619	0.714
	Strategy Two	CLS Pooling	True	NTXent	0.624	0.718
	Strategy Two	Mean-Max Pooling	True	NTXent	0.626	0.717
	Strategy Two	LSTM plus Attention Pooling	True	NTXent	0.628	0.725
xlm-roberta-large	Strategy Two	CLS Pooling	True	NTXent	0.626	0.738
	Strategy Two	Mean-Max Pooling	True	NTXent	0.625	0.736
	Strategy Two	LSTM plus Attention Pooling	True	NTXent	0.629	0.740
deberta-v3-large	Strategy Two	CLS Pooling	True	NTXent	0.647	0.771
	Strategy Two	Mean-Max Pooling	True	NTXent	0.645	0.773
	Strategy Two	LSTM plus Attention Pooling	True	NTXent	0.649	0.782
Multi model Fusion	—	—	—	—	0.654	0.785

Table 1: Experiment results for sub-task1 and sub-task2

reach a test accuracy of 0.654 on sub-task1 and a test Spear-man coefficient of 0.785 on sub-task2.

6 Conclusion

This paper propose a complex system with GLLRD strategy, contrastive loss, input data construction block, different pretrained models and different pooling head structure. It solves the problem of how to evaluate the quality of all possible fillers in cloze task. Experiments on SemEval task 7 datasets demonstrate that using our system can advance the normal cloze task models.

Acknowledgements

This research was supported by the PingAn Life Insurance. All the work stated in this paper was

conducted during the Semeval-2022 competition.

References

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 767–776. ACM.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Leyang Cui, Sijie Cheng, Yu Wu, and Yue Zhang. 2020. Does BERT solve commonsense task via commonsense knowledge? *CoRR*, abs/2008.03945.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

B. Ginsburg, I. Gitman, and Y. You. 2018. Large batch training of convolutional networks with layer-wise adaptive rate scaling.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Ayako Hoshino and Hiroshi Nakagawa. 2007. A cloze test authoring system and its automation. In *Advances in Web Based Learning - ICWL 2007, 6th International Conference, Edinburgh, UK, August 15-17, 2007, Revised Papers*, volume 4823 of *Lecture Notes in Computer Science*, pages 252–263. Springer.

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. *CoRR*, abs/1907.11692.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Michael Roth, Talita Anthonio, and Anna Sauer. SemEval-2022 Task 7: Identifying plausible clarifications of implicit and underspecified phrases in instructional texts. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*.

Yuqiang Xie, Yue Hu, Luxi Xing, Chunhui Wang, Yong Hu, Xiangpeng Wei, and Yajing Sun. 2020. Enhancing pre-trained language models by self-supervised learning for story cloze test. In *Knowledge Science, Engineering and Management - 13th International Conference, KSEM 2020, Hangzhou, China, August 28-30, 2020, Proceedings, Part I*, volume 12274 of *Lecture Notes in Computer Science*, pages 271–279. Springer.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. 2021. Revisiting few-sample BERT fine-tuning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.