IIE-NLP-Eyas at SemEval-2021 Task 4: Enhancing PLM for ReCAM with Special Tokens, Re-Ranking, Siamese Encoders and Back Translation

Yuqiang Xie Luxi Xing Wei Peng Yue Hu[§]
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{xieyuqiang,xingluxi,pengwei,huyue}@iie.ac.cn

Abstract

This paper introduces our systems for all three subtasks of SemEval-2021 Task 4: Reading Comprehension of Abstract Meaning. To help our model better represent and understand abstract concepts in natural language, we well-design many simple and effective approaches adapted to the backbone model (RoBERTa). Specifically, we formalize the subtasks into the multiple-choice question answering format and add special tokens to abstract concepts, then, the final prediction of QA is considered as the result of subtasks. Additionally, we employ many finetuning tricks to improve the performance. Experimental results show that our approach gains significant performance compared with the baseline systems. Our system[¶] achieves eighth rank (87.51%) and tenth rank (89.64%) on the official blind test set of subtask 1 and subtask 2 respectively.

1 Introduction

The computer's ability in understanding, representing, and expressing abstract meaning is a fundamental problem towards achieving true natural language understanding. SemEval-2021 Task 4: Reading Comprehension of Abstract Meaning (ReCAM) provides a well-formed benchmark that aims to study the machine's ability in representing and understanding abstract concepts (Zheng et al., 2021).

The Reading Comprehension of Abstract Meaning (ReCAM) task is divided into three subtasks, including Imperceptibility, Nonspecificity, and Interaction. Please refer to the task description paper (Zheng et al., 2021) for more details. To address the above challenges in ReCAM, we first formalize all subtasks as a type of multiple-choice Question Answering (QA) task like (Xing et al., 2020). Recently, the large Pre-trained Language Models (PLMs), such as GPT-2 (Radford et al., 2019), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), demonstrate their excellent ability in various natural language understanding tasks (Wang et al., 2018; Zellers et al., 2018, 2019). So, we employ the state-of-the-art PLM, RoBERTa, as our backbone model. Moreover, we design many simple and effective approaches to improve the performance of the backbone model, such as adding special tokens, sentence re-ranking, label smoothing and back translation.

This paper describes approaches for all subtasks developed by the IIE-NLP-Eyas Team (Natural Language Processing group of Institute of Information Engineering of the Chinese Academy of Sciences). Our contributions are summarized as the followings:

- We design many simple and effective approaches to improve the performance of the PLMs on all three subtasks, such as special tokens, sentence re-ranking, siamese encoders and back translation and label smoothing;
- Experiments demonstrate that our proposed methods achieve significant improvements compared with baselines and we obtain the 8th-place in subtask-1 and the 10th-place in subtask-2 on the final official evaluation.

2 Approaches

Since the format of the tasks in ReCAM is the same, we use the unified framework to address all tasks. The following is the detail of our methods.

Task Definition We first present the description of symbols which are used in this paper. Formally, suppose there are seven key elements in all subtasks, i.e. $\{D, Q, A_1, A_2, A_3, A_4, A_5\}$. We sup-

[§]Corresponding author.

[¶]Our Code is publicly available at https://github.com/indexfziq/IIE-NLP-Eyas-SemEval2021.

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 199–204
Bangkok, Thailand (online), August 5–6, 2021.

pose the D denotes the given article, the Q denotes the summary of the article with a *placeholder*, the A_* denotes the candidate abstract concepts for all subtasks to fill in the *placeholder*.

Multi-Choice Based Model The pre-trained language models have made a great contribution to MRC tasks. Recently, a significant milestone is the BERT (Devlin et al., 2019), which gets new state-of-the-art results on eleven natural language processing tasks. In this section, we present the description of the multi-choice based model which we use in all subtasks. Consider the BERT-style model RoBERTa's (Liu et al., 2019) stronger performance than BERT, we utilize it as our backbone model, which introduces more data and bigger models for better performance. A multiple-choice based QA model $\mathcal{M}$ consists of a PLM encoder and a task-specific classification layer which includes a feed-forward neural network $f(\cdot)$ and a softmax operation. For each pair of question-answer, the calculation of $\mathcal{M}$ is as follow:

$$score_i = \frac{exp(f(S_i))}{\sum_{i'} exp(f(S_{i'}))} \quad (1)$$

$$S_i = \text{PLM}([Q; A_i; D]) \quad (2)$$

where the $[\cdot]$ is the input constructed according to the instruction of PLMs, and the S_* is the final hidden state of the first token (<s>). For more details, we refer to the original work of PLMs (Liu et al., 2019). The candidate answer which owns a higher $score$ will be identified as the final prediction. The model $\mathcal{M}$ is trained end-to-end with the cross-entropy objective function.

Special Tokens Considering the great performance of special tokens in entity and relation extraction (Zhong and Chen, 2021), as well as of the prompt template on commonsense reasoning (Xing et al., 2020), we attach special tokens to highlight the semantic representation of candidate abstract concepts in the input layer. To help the PLMs represent and understand the abstract concept (i.e. option word in ReCAM tasks) in textual description (i.e. summary of the article in ReCAM task), we use <e> and </e> to add on both ends of the abstract concept, i.e. <e> abstract concept </e>. It is interesting that the special tokens are useful features contributing to most of the system's boost, and we have tried many other useful special tokens which will be discussed in section 4.

Sentence Ranking As the given passage is too long to be deal with the Pre-trained Language Models (PLMs), we consider refining the passage input by rearranging the order of the sentences in the passage. With this reorder process, the sentence, which is more critical to the question, can appear at the beginning of the passage. Although the passage's sequential information is sacrificed, we keep the more question-relevant information of the passage. Supposing the passage D contains N sentences, i.e., $D = \{W_1, W_2, ..., W_N\}$, where each sentence $W_n = \{t_1, t_2, ..., t_M\}$ including M tokens. We denote the given cloze-style question as Q. To rank the sentences in D, we resort BERT to compute the similarity score between each sentence, i.e. W_n, and Q following the algorithm in Zhang et al. (2020). After ranking, the sentences in D are sorted in descending order of similarity scores, and we can get a rearranged passage $\hat{D}$ as the passage input to the QA model. In the implement progress, $\hat{D}$ will be truncated to fit into the PLM encoder with our setting max length.

Siamese Encoders When exploring the dataset, we find that the complete question statement, representing the result statement after replacing the *placeholder* token with the candidate option, also contains the semantic information which can help to make the judgment about options. Based on the observation, we propose a siamese encoders based architecture to inject the additional complete question statement information while not influence the input with passage. On the other hand, it can be seen as introducing an auxiliary task to assist the main task. Specifically, the training of siamese encoder based architecture is as following:

$$l_i^1 = PLM([\hat{Q}_i])[0] \quad (3)$$

$$l_i^2 = PLM([Q; A_i; D])[0] \quad (4)$$

$$P^1(A_i|\hat{Q}) = \text{softmax}(f(l_i^1)) \quad (5)$$

$$P^2(A_i|D, Q) = \text{softmax}(f(l_i^2)) \quad (6)$$

where the $PLM(\cdot)$ stands for PLM encoder, $\hat{Q}_i$ is the complete question statement, i indicates the i-th candidate answer, $f(\cdot)$ is the feed forward network. To coordinate the two losses, we opt for an uncertainty loss (Kendall et al., 2018) to adjust it adaptively through $\sigma_{\{1,2\}}$ as: $\mathcal{L}(\theta, \sigma_1, \sigma_2) = \frac{1}{2\sigma_1^2}\mathcal{L}^1(\theta) + \frac{1}{2\sigma_2^2}\mathcal{L}^2(\theta) + \log\sigma_1^2\sigma_2^2$, where $\mathcal{L}^{\{1,2\}}$ are the cross-entropy loss between the model prediction $P^{\{1,2\}}$ and the ground truth label respectively.

Back Translation Generally speaking, more successful neural networks require a large number of parameters, often in the millions. In order to make the neural network implements correctly, a lot of data is needed for training, but in actual situations, there is not as much data as we thought. The role of data augmentation includes two aspects. One is to increase the amount of training data and improve the generalization ability of the model. The other is to increase the noise data and improve the robustness of the model. A large number of the works (Buslaev et al., 2018; Bloice et al., 2019; Chen et al., 2020; Cubuk et al., 2020; Sato et al., 2018; Zhu et al., 2020) consider the data augmentation to make better performances. In the field of computer vision, a lot of work (Buslaev et al., 2018; Bloice et al., 2019; Chen et al., 2020; Cubuk et al., 2020) uses existing data to perform operations, such as flipping, translation or rotation, to create more data, so that neural networks have better generalization effects. Adding Gaussian distribution to text processing (Sato et al., 2018) can also achieve the effect of data augmentation. Besides, some works (Miyato et al., 2017; Zhu et al., 2020) utilize the adversarial training methods to do the data augmentation. For convenience and simplicity, we adopt the back translation (Sennrich et al., 2016) to increase the amount of training data, which is used to construct pseudo parallel corpus in unsupervised machine translation (Lample et al., 2018). Specifically, we use the Google API[†] to translate the passage into French, and then translate the translation into English in turn. The pseudo parallel corpus can be obtained as:

$$\{D'\} = bkt(\{D\}) \quad (7)$$

where $\{D'\}$ means the translated English corpus that we used as data agument, bkt is back translation.

As for the question, given the existence of the special character *placeholder*, forced translation may result in grammatical errors and semantic gaps. Therefore, the questions and options will be kept original. After getting the pseudo parallel corpus, we train our model with the training data together with the cross-entropy loss function.

Label Smoothing Furthermore, for improving the generalization ability of the model trained on sole task and prevent the overconfidence of model, we consider training model with label smoothing (Miller et al., 1996; Pereyra et al., 2017). Label smoothing can maintain uncertainty over the label space during training. When training with label smoothing, for classification tasks, the hard one-hot label distribution is replaced with a softened label distribution through a smoothing value α, which is a hyperparameter. Specifically, for hard one-hot label distribution, the target category's probability will be assigned to 1.0 and others are 0.0. Label smoothing will soften the label distribution by modifying the probability distribution with a discount. Then, the target category's probability will be $1-\alpha$, and the probabilities of the rest categories are $\frac{\alpha}{\mathcal{K}-1}$, where $\mathcal{K}$ is the number of task categories. In our experiments, we set the smoothing value $\alpha = 0.1$.

Subtask	Train	Trail	Dev	Test
Imperceptibility	3227	1000	837	2025
Nonspecificility	3318	1000	851	2017

Table 1: Data scale of each subtask.

Hyper-parameter	Value
LR	{1e-5, 2e-5}
Batch size	{16, 32}
Gradient norm	1.0
Warm-up	{0.1, 1, 2}
Max. input length (# subwords)	200
Epochs	[3, 10]

Table 2: Hyper-parameters of our approach.

3 Experiments and Results

3.1 Experimental Setup

In all subtasks, the scale of each task is shown in Table 1. We train the model on training data and the related pseudo data generated by back translation, then select hyper-parameters based on the best performing model on the dev set, and then report results on the test set.

Our system is implemented with PyTorch and we use the PyTorch version of the pre-trained language models[‡]. We employ RoBERTa (Liu et al., 2019) large model as our PLM encoder in Equation 2. The Adam optimizer (Kingma and Ba, 2014) is used to fine-tune the model. We introduce the detailed setup of the best model on the development dataset. For subtask-1 and subtask-2, the hyper-parameters are shown in Table 2.

[†]The web page is available at https://translate.google.com

[‡]https://github.com/huggingface/transformers

Models	Trial Acc.	Dev Acc.
ROBERTA$_{\text{LARGE}}$(Liu et al., 2019)	85.85	82.12
(1) w/ special tokens	**87.81**	**87.69**
(2) w/ sentence ranking	86.54	83.52
(3) w/ label smoothing	86.88	85.85
(4) w/ siamese encoders	86.62	83.22
(5) w/ back translation	87.23	84.32
Our Approach	**87.81**	**87.69**

Table 3: The results of our system on subtask-1. Our approach is the final, stable and best model: ROBERTA$_{\text{LARGE}}$ with special tokens. We finally obtain 87.51 Acc. on the official blind test set.

Models	Trial Acc.	Dev Acc.
ROBERTA$_{\text{LARGE}}$(Liu et al., 2019)	**88.51**	85.93
(1) w/ special tokens	87.47	88.98
(2) w/ sentence ranking	87.29	86.84
(3) w/ label smoothing	87.67	87.08
(4) w/ siamese encoders	87.34	86.18
(5) w/ back translation	88.41	87.54
Our Approach	87.10	**89.54**

Table 4: The results of our system on subtask-2. Our approach is the final, stable and best model: ROBERTA$_{\text{LARGE}}$ with special tokens and label smoothing. We finally obtain 89.64 Acc. on the official blind test set.

3.2 Evaluation Results

Imperceptibility From Table 3, we can see the results of our system on subtask-1 of ReCAM. Compared to the backbone model RoBERTa large model, our methods achieve significant improvements. It is interesting that the special token is the most helpful part for the Imperceptibility subtask.

Nonspecificility Table 4 summarizes the results of our approachs on subtask-2 of ReCAM. In Nonspecificility subtask, the model with special tokens and label smoothing performs best. Compared to the backbone model ROBERTA$_{\text{LARGE}}$, all our methods achieve better performance.

Interaction We also perform subtask-3 of ReCAM, Interaction, which aims to provide more insights into the relationship of the two views on abstractness. In this task, we test the performance of our system that is trained on one definition and evaluated on the other. The results of our system's performance on Imperceptibility and Nonspecificility subtasks which is shown in Table 5. We can find that our model is relatively robust for different abstract concepts.

Trained on	Tested on	Test Acc.
Subtask-1	Subtask-1	87.51
Subtask-1	Subtask-2	84.13
Subtask-2	Subtask-2	89.64
Subtask-2	Subtask-1	81.09

Table 5: The results of our approach on subtask-3.

Special Token	Trial Acc.	Dev Acc.
<e> </e>	88.01	**87.10**
<#> </#>	**88.63**	86.93
<$> </$>	88.12	86.26
# /#	87.34	85.89
$ /$	87.73	86.13
N/A	86.23	83.12

Table 6: The results of models with different special tokens on subtask-1.

4 Analysis and Discussion

4.1 Ablation Study

In this part, we perform an ablation study of our approaches (special tokens, sentence re-ranking, label smoothing, siamese encoders and back translation).

Table 3 and 4 shows that our proposed methods help the backbone model better represent and understand the abstract concepts. Note that the special tokens bring the PLMs with the best improvements in both subtask-1 and subtask-2. It is possible that the special tokens teach the model to focus on the abstract concept in a stronger manner. Moreover, other common tricks bring with little improvements.

4.2 Discussion of Special Tokens

We also search for the best special tokens for ReCAM on the dev set of subtask-1. `e` stands for the word `entity`. `#` and `$` are common special tokens for NLP downstream applications.

As shown in Table 6, `<e> </e>` enhance the representations of abstract concepts best of all. `#` and `$` work well. In addition, the `<>` and `</>` could be helpful for PLMs to pay attention to the abstract concepts. Moreover, it is interesting that each special token helps PLMs choose the right abstract concepts which are submerged in long sequential tokens (including article and summary). This result strengthen the point that special tokens can enhance the representation of abstract concepts in PLM based approaches.

5 Conclusion

In this paper, we design many simple and effective approaches to improve the performance of the PLMs on all three subtasks. Experiments demonstrate that the proposed methods achieve significant improvement compared with the PLMs baseline and we obtain the eighth-place in subtask-1 and tenth-place in subtask-2 on the final official evaluation. Moreover, we show that special tokens are useful features contributing to most of the system's boost, which work well in enhancing PLMs for representating and understanding abstract concepts.

Acknowledgements

We thank the anonymous reviewers for their insightful feedback. This work has been supported by the National Key Research and Development Program of China under Grant No.Y750211101.

References

Marcus D. Bloice, Peter M. Roth, and Andreas Holzinger. 2019. Biomedical image augmentation using augmentor. *Bioinform.*, 35(21):4522–4524.

Alexander V. Buslaev, Alex Parinov, Eugene Khvedchenya, Vladimir I. Iglovikov, and Alexandr A. Kalinin. 2018. Albumentations: fast and flexible image augmentations. *CoRR*, abs/1809.06839.

Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. 2020. Gridmask data augmentation. *CoRR*, abs/2001.04086.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pages 3008–3017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7482–7491.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

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.

David J. Miller, Ajit V. Rao, Kenneth Rose, and Allen Gersho. 1996. A global optimization technique for statistical classifier design. *IEEE Trans. Signal Process.*, 44(12):3108–3122.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*.

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

Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4323–4330.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. 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*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Luxi Xing, Yuqiang Xie, Yue Hu, and Wei Peng. 2020. IIE-NLP-NUT at SemEval-2020 task 4: Guiding plm with prompt template reconstruction strategy for comve. In *SemEval@COLING*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore*, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Boyuan Zheng, Xiaoyu Yang, Yuping Ruan, Quan Liu, Zhen-Hua Ling, Si Wei, and Xiaodan Zhu. 2021. SemEval-2021 task 4: Reading comprehension of abstract meaning. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *North American Association for Computational Linguistics (NAACL)*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.