CLER: Cross-task Learning with Expert Representation to Generalize Reading and Understanding

Takumi Takahashi *
takahashi.takumi@fujixerox.co.jp

Motoki Taniguchi *
motoki.taniguchi@fujixerox.co.jp

Tomoki Taniguchi
Taniguchi.Tomoki@fujixerox.co.jp

Tomoko Ohkuma
Ohkuma.Tomoko@fujixerox.co.jp

Fuji Xerox Co., Ltd.

Abstract

This paper describes our model for the reading comprehension task of the MRQA shared task. We propose CLER, which stands for **C**ross-task **L**earning with **E**xpert **R**epresentation for the generalization of reading and understanding. To generalize its capabilities, the proposed model is composed of three key ideas: multi-task learning, mixture of experts, and ensemble. In-domain datasets are used to train and validate our model, and other out-of-domain datasets are used to validate the generalization of our model's performances. In a submission run result, the proposed model achieved an average F1 score of 66.1 % in the out-of-domain setting, which is a 4.3 percentage point improvement over the official BERT baseline model.

Dataset	Size	Context	Question
SQuAD	96K	wikipedia	crowd
NewsQA	78K	newswire	crowd
TriviaQA	69K	snippets	quiz
SearchQA	133K	snippets	quiz
HotpotQA	78K	wikipedia	crowd
NaturalQuestions	116K	wikipedia	crowd
DROP	1,503	wikipedia	crowd
RACE	674	exam	handcraft
BioASQ	1,504	biomedical	handcraft
TextbookQA	1,503	textbook	handcraft
RelationExtraction	2,948	wikipedia	KB
DuoRC	1,501	plot	crowd

Table 1: Characteristics of released datasets for the MRQA shared task. The top part of the table indicates in-domain datasets to train and validate the model, and the bottom part of the table indicates unveiled out-of-domain datasets to validate the generalization of the trained model.

1 Introduction

Reading comprehension (RC) tasks are important to measure machines' capabilities of reading and understanding. Given a question and context, a typical extractive RC task aims to automatically extract an appropriate answer from the given context.

A large number of datasets for RC tasks, which contains various types of context, such as Wikipedia article (Rajpurkar et al., 2016; Yang et al., 2018; Kwiatkowski et al., 2019), newswire (Trischler et al., 2017), and web snipets (Dunn et al., 2017; Joshi et al., 2017), have recently been published. Similarly, many types of RC task, such as multiple passage (Dunn et al., 2017; Joshi et al., 2017), multi-hop reasoning (Yang et al., 2018; Welbl et al., 2018), dialog (Choi et al., 2018; Reddy et al., 2019) and commonsense reasoning (Ostermann et al., 2018; Talmor et al., 2019), are contained in recently published datasets.

To assess the performance of an RC model on such datasets, basically, we have to train the model on the target domain. This solution requires the same domain dataset as the target domain to appropriately train the model. However, it is difficult to collect the same domain dataset whenever we train a model for an RC task.

To overcome this problem, transfer learning can be applied to create a general model, but there have been few works on this (Chung et al., 2018; Talmor and Berant, 2019; Sun et al., 2019). During training on the source dataset, the model should be generalized to prevent overfitting to the particular domain. In other words, the model should be able to deal with examples on the target domain (i.e., out-of-domain) well.

The MRQA shared task aims to measure generalization capability for RC tasks. The shared task released six-domain datasets (Rajpurkar et al., 2016; Trischler et al., 2017; Joshi et al., 2017; Dunn et al., 2017; Yang et al., 2018; Kwiatkowski et al., 2019) to train and vali-

*Authors contributed equally

Proceedings of the Second Workshop on Machine Reading for Question Answering, pages 183–190
Hong Kong, China, November 4, 2019. ©2019 Association for Computational Linguistics

date the model as in-domain settings, and unveiled six out of the twelve test datasets [1] (Dua et al., 2019; Lai et al., 2017; Kembhavi et al., 2017; Levy et al., 2017; Saha et al., 2018) to validate the trained model as out-of-domain settings. The characteristics of released datasets are shown in Table 1. The goal of this competition is to demonstrate high performances on out-of-domain datasets (the bottom part of Table 1 and additionally unseen test datasets) by the trained model which only utilizes in-domain datasets (the top part of Table 1).

In this paper, we propose **CLER**, which stands for **C**ross-task **L**earning with **E**xpert **R**epresentation. CLER is based on BERT (Devlin et al., 2019), which has recently shown great success as a large-scale language model. The proposed model is composed of three concepts; multi-task learning, mixture of experts (MoE), and ensemble.

Our first motivation to employ multi-task learning is inspired by MT-DNN (Liu et al., 2019a). MT-DNN is based on BERT as a shared layer and is trained on four tasks: single-sentence classification, pairwise text similarity, pairwise text classification, and pairwise ranking. In particular, natural language inference (NLI) as a pairwise sentence classification task is related to RC tasks, even in four tasks. Therefore, we train the proposed model for RC and NLI tasks in a multi-task setting.

Our second motivation to employ MoE is inspired by Guo et al. (2018). They demonstrated the effectiveness of the MoE architecture for transfer learning in sentiment analysis and part-of-speech tagging tasks. MoE basically has different neural networks called "experts" and divides a single task into several subtasks so that each subtask is assigned to one expert. Here, we assume that each subtask corresponds to each domain in in-domain settings. Moreover, in MoE, unseen domains (i.e., out-of-domain) are represented as a combination of several domains, such as SQuAD, TriviaQA, and HotpotQA. Therefore, we expect that MoE can deal with examples in any domain well.

Finally, we employ an ensemble to enhance the performance of the proposed model. Because ensemble models have shown superior performances over than single ones (Seo et al., 2016; Devlin et al., 2019), we introduce an ensemble mechanism to improve performance.

The contributions of this paper are as follows:

- We propose a BERT-based model with multi-task learning and mixture of experts called **CLER**.
- We demonstrate that our model has better performances than the official BERT baseline model in both in-domain and out-of-domain settings.

2 Related works

RC models: The state-of-the-art in RC tasks has been rapidly advanced by neural models (Seo et al., 2016; Yu et al., 2018; Devlin et al., 2019). In particular, BERT (Devlin et al., 2019) significantly improves the performance of a wide range of natural language understanding tasks, including RC tasks. BERT is designed to pre-train contextual representations from unlabeled text and fine-tune for downstream tasks. By leveraging large amounts of unlabeled data, BERT can obtain rich contextual representations.

Multi-task learning: Multi-task learning (Caruana, 1997) is a widely used technique in which a model is trained on data from multiple tasks. Multi-task learning provides the model a regularization effect to alleviate overfitting to a specific task, thus enabling universal representations to be learned across tasks. Liu et al. (2019a) proposed the multi-task deep neural network (MT-DNN) based on the BERT model. Similar to the original BERT model, MT-DNN is pre-trained as a language model for learning contextual representations. In the fine-tuning phase, MT-DNN uses multi-task learning instead of training on only a specific task.

Mixture-of-Experts : Guo et al. (2018) introduced the mixture-of-experts (MoE) (Jacobs et al., 1991) approach for unsupervised domain adaptation from multiple sources. MoE is composed of different neural networks, i.e., experts. In the original MoE, a single task is divided into subtasks, and each expert learns to handle a certain subtask. Guo et al. (2018) assumes that different source domains are aligned to different sub-spaces of the target domain.

3 Model

For generalization to RC tasks, we propose CLER, which is based on BERT (Devlin et al., 2019) and

[1] BioASQ: http://bioasq.org/

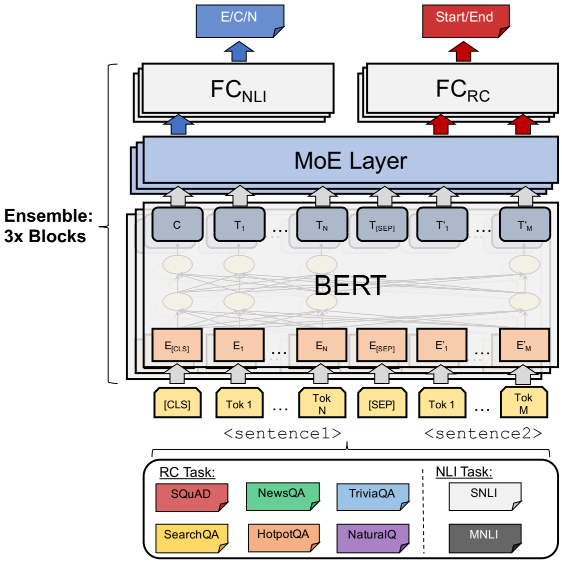

Figure 1: Overview of the proposed model called **CLER**. Each block in the single model consists of BERT, MoE, and FC layers. All three blocks are aggregated into an ensemble CLER. Each block is trained with a different seed.

several other techniques. An overview of the proposed model is illustrated in Figure 1. The core concepts behind our model are multi-task learning, mixture of experts (MoE), and the ensemble mechanism. During training, MoE learns the relationship between domains regardless of the type of task, while the model is trained on RC and NLI tasks simultaneously. We refer to this series of training procedures that trains the model with different experts on two types of task as **cross-task learning**.

3.1 BERT-based model

We utilize BERT$_{\text{LARGE}}$ to encode a pair of sentences composed as `[CLS] <sentence1> [SEP] <sentence2>`. BERT$_{\text{LARGE}}$, which consists of 24 transformer blocks, has already been pre-trained using BooksCorpus (Zhu et al., 2015) and English Wikipedia. For an RC task, the given question and context are set to `<sentence1>` and `<sentence2>`, respectively. Similarly, for an NLI task, the given premise and hypothesis are set to `<sentence1>` and `<sentence2>`, respectively. `[CLS]` and `[SEP]` are special tokens prepared by the default function of BERT. The given pair of sentences is tokenized as a wordpiece token with a sequence length of up to $\tilde{L} = 512$. Finally, all tokens are fed into the MoE layer.

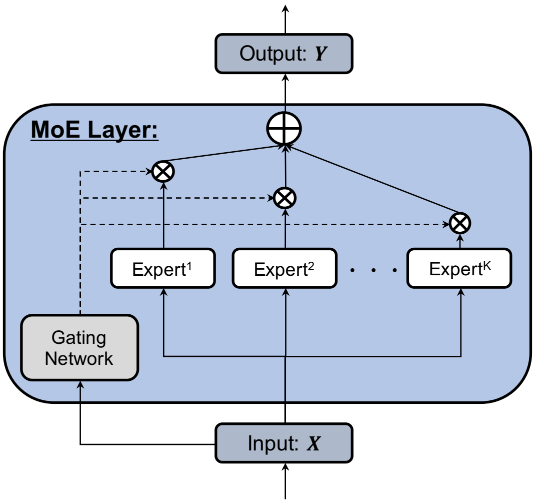

Figure 2: Architecture of the MoE layer. $\otimes$ represents the multiplication operator, and $\oplus$ represents the summation operator.

3.2 Mixture of Experts

To explicitly capture the representation between domains, we introduce a mixture of experts (MoE) (Jacobs et al., 1991) layer after encoding the representation over BERT. As illustrated in Figure 2, MoE is composed of K parts in the expert layer to encode the input representation and a gating network to classify the input representation into the local experts. Intuitively, we expect that each expert is able to interpret domain-wise representations.

Formally, given the representation $\boldsymbol{X} \in \mathbb{R}^{d \times L}$, where d is the number of dimensions of the output of BERT and L indicates the number of input tokens, the equation for output $\boldsymbol{Y} \in \mathbb{R}^{d \times L}$ can be written as follows:

$$\boldsymbol{Y} = \sum_{i=1}^{K} G(\boldsymbol{X})_i E_i(\boldsymbol{X}) \tag{1}$$

where $G(\boldsymbol{x})_i$ indicates the output probability of the i-th expert via the gating network, $E_i(\boldsymbol{x})$ indicates the output representation via the i-th expert layer, and K is the total number of experts.

Here, we give the equations of the gating network $G(\cdot)$ as follows:

$$G(\boldsymbol{X}) = softmax(\boldsymbol{W}_g \boldsymbol{h} + \boldsymbol{b}_g), \tag{2}$$

$$\boldsymbol{h} = [\overrightarrow{h_L}; \overleftarrow{h_1}], \tag{3}$$

$$\overrightarrow{h_L} = \overrightarrow{\text{GRU}}(\boldsymbol{X}), \overleftarrow{h_1} = \overleftarrow{\text{GRU}}(\boldsymbol{X}), \tag{4}$$

where $\overrightarrow{\text{GRU}}$ and $\overleftarrow{\text{GRU}}$ correspond to a forward GRU and backward GRU, respectively, $\boldsymbol{W}_g$ is a weight matrix, $\boldsymbol{b}_g$ is a bias vector, ; indicates a concatenation operator, and L is the number of given tokens. Note that each GRU only outputs the final hidden state vector in Equation 4.

Then, we give the equation of the i-th expert layer $E(\cdot)$ as follows:

$$E_i(\boldsymbol{X}) = \boldsymbol{W}_i\boldsymbol{X} + \boldsymbol{b}_i \tag{5}$$

where $\boldsymbol{W}_i$ is the i-th weight matrix, and $\boldsymbol{b}_i$ is the i-th bias vector.

As mentioned above, each expert has a different weight matrix and bias vector, and the gating network classifies an input example into local experts. Therefore, all experts are able to interpret the input representation with respect to any domain, even if it is unseen in the source domain.

3.3 Multi-task Learning

According to Liu et al. (2019a), multi-task learning is effective for improving models on several NLP tasks. In particular, NLI tasks are related to RC tasks and even several NLP tasks. Therefore, we employ the multi-task learning approach on RC and NLI tasks to enhance the generalization of our model.

BERT-encoder and MoE layer correspond to a shared layer, and both $\mathrm{FC_{RC}}$ and $\mathrm{FC_{NLI}}$, which indicate fully connected layers, are task-specific layers in our multi-task setting. For $\mathrm{FC_{RC}}$ at prediction time, given the representation of all tokens via the MoE layer, $\mathrm{FC_{RC}}$ outputs the span with the maximum logits across all tokens. Specifically, two types of $\mathrm{FC_{RC}}$ layer, which are span predictors for the start and end position, estimate the span with the start and end position, individually. For $\mathrm{FC_{NLI}}$ at prediction time, given the representation of the first token via the MoE layer corresponding to the `[CLS]` token, $\mathrm{FC_{NLI}}$ outputs a predicted class out of *entailment*, *neutral*, and *contradiction*.

Loss Function

Finally, we minimize the loss function with the multi-task setting as follows:

$$\mathcal{L} = \lambda\mathcal{L}_{RC} + (1-\lambda)\mathcal{L}_{NLI} + \mathcal{L}_{importance} \tag{6}$$

where $\mathcal{L}_{RC}$ is a negative log likelihood loss for RC tasks, $\mathcal{L}_{NLI}$ is a cross entropy loss for NLI tasks, $\mathcal{L}_{importance}$ is an importance loss, and λ is a weight hyperparameter.

According to Shazeer et al. (2017), we employ an importance loss $\mathcal{L}_{importance}$ to avoid the local minimum. This loss function penalizes some experts that frequently take a large probability via the gating network in any domain. Let us denote the importance loss as follows:

$$\mathcal{L}_{importance} = w_{importance} CV(I(Z))^2 \tag{7}$$

$$I(Z) = \sum_{z \in Z} G(z) \tag{8}$$

where Z represents all samples in the given mini-batch, $CV(\cdot)$ is a coefficient of variation, and $w_{importance}$ is a weight hyperparameter.

3.4 Ensemble

To further enhance the generalization of our model, we employ an ensemble mechanism. The ensemble is only applied at test time.

At test time, we feed examples of RC tasks into our models, which are trained with different seeds, independently. We integrate the logits via $\mathrm{FC_{RC}}$ into a merged logit as follows:

$$\boldsymbol{m}_s = \sum_{j=1}^{J} \boldsymbol{o}_s^j, \qquad \boldsymbol{m}_e = \sum_{j=1}^{J} \boldsymbol{o}_e^j, \tag{9}$$

where $\boldsymbol{o}_s^j \in \mathbb{R}^L$ and $\boldsymbol{o}_e^j \in \mathbb{R}^L$ correspond to the logits of our j-th model for the start span and end span, respectively, and J is the total number of models in the ensemble. Finally, we take the span with the maximum logits over $\boldsymbol{m}_s$ and $\boldsymbol{m}_e$.

4 Experiments

4.1 Datasets

Datasets for RC Tasks

MRQA shared task organizers released six types of train and development dataset to train and validate the model for generalization. Additionally, six out of the twelve types of out-of-domain dataset were unveiled to only validate the trained model.

We randomly sampled examples to make the **Test** set from the official train dataset. Note that **Train**, which was created from the official train dataset but is not the same as the official one, does not contain the same examples as in **Test**. The development dataset **Dev.** was used as the same for the official development set. The statistics of the datasets are listed in Table 2.

Dataset	**Train**	**Dev.**	**Test**
SQuAD	76,079	10,507	10,509
NewsQA	69,947	4,212	4,213
TriviaQA	53,902	7,785	7,786
SearchQA	100,403	16,980	16,981
HotpotQA	67,010	5,904	5,902
NaturalQuestions	91,234	12,836	12,837
DROP	-	1,503	-
RACE	-	674	-
BioASQ	-	1,504	-
TextbookQA	-	1,503	-
RelationExtraction	-	2,948	-
DuoRC	-	1,501	-

Table 2: Statistics of datasets for RC tasks. The top part of the table indicates in-domain datasets to train and validate the model, and the bottom part of the table indicates unveiled out-of-domain datasets to only validate the trained model.

Dataset	**Train**	**Dev.**
SNLI	550,152	10,000
FICTION	77,348	2,000
GOVERNMENT	77,350	2,000
SLATE	77,306	2,000
TELEPHONE	83,348	2,000
TRAVEL	77,350	2,000

Table 3: Statistics of datasets for NLI tasks. The bottom part of the table indicates genres in the MNLI dataset.

At training time, we took only 75 K examples from each dataset if the total number of examples in the dataset was larger than 75 K. Otherwise, we took all examples in the dataset.

Datasets for NLI Tasks

We introduce two types of NLI datasets to train our model with multi-task learning: SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018). The statistics of these datasets are listed in Table 3.

At training time, the number of examples in each dataset corresponded to the number of examples in the RC task dataset. Specifically, the numbers of examples on SNLI, FICTION, GOVERNMENT, SLATE, TELEPHONE, and TRAVEL were the same as those of SQuAD, NewsQA, TriviaQA, SearchQA, HotpotQA, and NaturalQuestions, respectively.

4.2 Experimental Setup

All of our implementations followed the settings described in this section.

We used the BERT$_{\text{LARGE}}$ model for all of our implementations. For the MoE layer, the number of experts was set to 12. We set the hidden unit sizes of the GRU layer and the hidden unit sizes of each expert to 512 and 1024, respectively. For the ensemble model, we trained three models independently with different seeds. The best model of the three evaluated on the out-of-domain development set was chosen as a single model.

We used Adam with a learning rate of 3e-5 to optimize the model. We fine-tuned the model for 2 epochs with a batch size of 24. During training, λ and $w_{importance}$ were set to 0.5 and 0.1, respectively.

Two types of metrics, exact match (EM) and partial match (F1), were employed in the MRQA shared task. EM was 1 if the predicted answer was perfectly the same as the gold answer, but otherwise it was 0. For F1, we calculated the overlap rate between the predicted answer and the gold answer, so the maximum F1 score is 1.

4.3 Comparison Models

As baseline models, we referred to the official evaluation results based on BERT$_{\text{BASE}}$ and BERT$_{\text{LARGE}}$. To fairly compare the baseline and our models, we prepared BERT$_{\text{STL}}$, which is composed of only the BERT-encoder and FC$_{\text{RC}}$ with the same settings of our models. BERT$_{\text{STL}}$ is different from BERT$_{\text{LARGE}}$ with respect to the hyperparameter of scheduling (t_total in Pytorch implementation). Note that BERT$_{\text{STL}}$ does not employ both multi-task learning and ensemble.

We also prepared BERT$_{\text{MTL}}$ excluding the MoE layer from CLER, as illustrated in Figure 1, to assess the effectiveness of multi-task learning.

4.4 Results

In-domain Evaluation

We evaluated all models on the in-domain development set. Table 4 summarizes the results on the in-domain development set.

CLER with the ensemble setting consistently demonstrated superior performances on all datasets. Also, the multi-task learning (BERT$_{\text{MTL}}$) effectively improved overall performances. However, MoE could not improve the performances compared with BERT$_{\text{MTL}}$ on in-domain datasets.

Out-of-domain Evaluation

We also evaluated all models on the out-of-domain development set. Table 5 summarizes the evaluation results for out-of-domain.

Model	SQuAD (EM/F1)	NewsQA (EM/F1)	TriviaQA (EM/F1)	SearchQA (EM/F1)	HotpotQA (EM/F1)	NaturalQuestions (EM/F1)	Average (EM/F1)
BERT$_{\mathrm{BASE}}$	78.5/86.7	50.8/66.8	65.6/71.6	69.5/76.7	59.8/76.6	65.4/77.4	64.9/76.0
BERT$_{\mathrm{LARGE}}$	80.3/88.4	49.6/66.3	68.2/74.7	71.8/79.0	62.4/79.0	67.9/79.8	66.7/77.9
BERT$_{\mathrm{STL}}$	83.3/90.5	51.5/67.4	68.5/74.3	72.2/79.3	63.9/80.1	67.7/79.7	67.9/78.5
BERT$_{\mathrm{MTL}}$	84.6/91.4	54.1/69.4	70.5/76.0	72.6/79.5	63.9/80.0	67.9/79.5	68.9/79.3
CLER (Single)	84.9/91.6	54.3/69.4	69.9/75.6	72.2/79.0	63.5/79.8	68.1/79.8	68.8/79.2
CLER (Ensemble)	**85.5/91.9**	**55.7/70.5**	**71.8/77.4**	**73.7/80.5**	**64.9/80.9**	**68.5/80.1**	**70.0/80.2**

Table 4: Results on the in-domain development set. Bold values indicate the best scores overall, and the underlined values indicate the best scores for each single model. BERT$_{\mathrm{STL}}$ is a single-task learning model composed of only a BERT-encoder and FC$_{\mathrm{RC}}$ based on our reimplementation. BERT$_{\mathrm{MTL}}$ is a multi-task learning model excluding the MoE layer from CLER.

Model	DROP (EM/F1)	RACE (EM/F1)	BioASQ (EM/F1)	TextbookQA (EM/F1)	RelationExtraction (EM/F1)	DuoRC (EM/F1)	Average (EM/F1)
BERT$_{\mathrm{BASE}}$	25.7/34.5	30.4/41.4	47.1/62.7	44.9/53.9	72.6/83.8	44.8/54.6	44.3/55.2
BERT$_{\mathrm{LARGE}}$	34.6/43.8	31.3/42.5	51.9/66.8	47.4/55.7	72.7/85.2	46.8/58.0	47.5/58.7
BERT$_{\mathrm{STL}}$	38.5/47.3	**33.7**/45.7	**53.9**/**69.6**	48.0/56.6	76.4/86.7	46.9/57.2	49.6/60.5
BERT$_{\mathrm{MTL}}$	37.9/46.8	30.4/44.4	53.5/69.0	49.8/58.9	76.9/87.0	51.4/60.8	50.0/61.2
CLER (Single)	39.3/47.8	32.3/**46.6**	52.8/67.4	51.4/61.0	76.3/87.0	51.8/61.8	50.7/62.0
CLER (Ensemble)	**40.2/49.4**	32.2/46.2	52.1/68.4	**52.6/62.3**	**77.3/87.7**	**52.2/61.9**	**51.1/62.7**

Table 5: Results on the out-of-domain development set. Bold values indicate the best scores overall, and the underlined values indicate the best scores for each single model. BERT$_{\mathrm{STL}}$ and BERT$_{\mathrm{MTL}}$ are the same as in Table 4.

Model	**Dev.** (EM/F1)	**Test** (EM/F1)	**Average** (EM/F1)
BERT$_{\mathrm{BASE}}$	43.9/54.6	47.2/62.4	45.5/58.5
BERT$_{\mathrm{LARGE}}$	45.7/57.4	50.7/66.1	48.2/61.8
CLER (Ensemble)	**51.1/62.5**	**53.8/69.7**	**52.4/66.1**

Table 6: Results of submission run. BERT$_{\mathrm{BASE}}$ and BERT$_{\mathrm{LARGE}}$ are the MRQA official baseline models. Bold values indicate the best scores overall.

Overall, the performances of our model were improved compared to the official baseline models. It was observed that CLER drastically improved the EM and F1 scores compared with baseline models on TextbookQA and DuoRC. Moreover, the multi-task learning improved the average F1 score (+0.7 pt) compared with BERT$_{\mathrm{STL}}$, and the MoE layer further improved the average F1 score (+0.8 pt) compared with BERT$_{\mathrm{MTL}}$. This suggests that both the multi-task learning and MoE are effective for improving generalization for RC tasks.

4.5 Submission Run

For the submission run, 6-domain datasets for the development set and additional 6-domain datasets for the test set were used to evaluate the submitted models. All datasets for the submission run were consistently out-of-domain settings.

Table 6 summarizes the submission run results. CLER drastically improved the performances compared with the official baseline models. We finally ranked 6th of all participants.

5 Conclusion

In this paper, we proposed a BERT-based model with multi-task learning and mixture of experts (MoE) called CLER. To enhance generalization for RC tasks, we introduced an MoE layer and the multi-task learning approach. We also applied an ensemble mechanism to CLER to further improve its performances. Experimental results showed that CLER drastically improved EM and F1 scores compared with the official BERT baseline models.

In future work, we will replace the BERT-encoder with a more powerful model, such as XLNet (Yang et al., 2019) or RoBERTa (Liu et al., 2019b), which have recently achieved state-of-the-art performances on natural language understanding benchmarks. We will also attempt other training strategies, such as question generation, to automatically augment the training dataset.

Acknowledgement

Computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

Yu-An Chung, Hung-Yi Lee, and James Glass. 2018. Supervised and unsupervised transfer learning for question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1585–1594.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Jiang Guo, Darsh Shah, and Regina Barzilay. 2018. Multi-source domain adaptation with mixture of experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4694–4703.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, Geoffrey E Hinton, et al. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1601–1611.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4999–5007.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

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

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018. MCScript: A novel dataset for assessing machine comprehension using script knowledge. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2019. Improving machine reading comprehension with general reading strategies. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2633–2643.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.