Pingan Smart Health and SJTU at COIN - Shared Task: utilizing Pre-trained Language Models and Common-sense Knowledge in Machine Reading Tasks

Xiepeng Li[1*] **Zhexi Zhang**[2*] **Wei Zhu**[1*†] **Zheng Li**[1] **Yuan Ni**[1]
Peng Gao[1] **Junchi Yan**[2] **Guotong Xie**[1]
[1] Pingan Health Tech, Shanghai, China
[2] Shanghai Jiaotong University, Shanghai, China

Abstract

To solve the shared tasks of COIN: COmmonsense INference in Natural Language Processing) Workshop in EMNLP-IJCNLP 2019, we need explore the impact of knowledge representation in modeling commonsense knowledge to boost performance of machine reading comprehension beyond simple text matching. There are two approaches to represent knowledge in the low-dimensional space. The first is to leverage large-scale unsupervised text corpus to train fixed or contextual language representations. The second approach is to explicitly express knowledge into a knowledge graph (KG), and then fit a model to represent the facts in the KG. We have experimented both (a) improving the fine-tuning of pre-trained language models on a task with a small dataset size, by leveraging datasets of similar tasks; and (b) incorporating the distributional representations of a KG onto the representations of pre-trained language models, via simply concatenation or multi-head attention. We find out that: (a) for task 1, first fine-tuning on larger datasets like RACE (Lai et al., 2017) and SWAG (Zellers et al., 2018), and then fine-tuning on the target task improve the performance significantly; (b) for task 2, we find out the incorporating a KG of commonsense knowledge, WordNet (Miller, 1995) into the Bert model (Devlin et al., 2018) is helpful, however, it will hurts the performace of XLNET (Yang et al., 2019), a more powerful pre-trained model. Our approaches achieve the state-of-the-art results on both shared task's official test data, outperforming all the other submissions.

1 Introduction

Machine reading comprehension (MRC) tasks have always been the most studied tasks in the field of natural language understanding. Common forms of reading comprehension tasks involve question answer (QA) , cloze-style and multiple-choice questions. Many models have achieved excellent results on MRC datasets such as (Rajpurkar et al., 2016; Nguyen et al., 2016; Lai et al., 2017; Zhang et al., 2018a). However, Kaushik and Lipton (2018) demonstrate that most questions in previous MRC tasks can be answered by simply matching the patterns in the textual level even with passage or question only, but existing models perform badly on questions that require incorporating knowledge in more sophisticated ways. In contrast, human beings can easily reason with knowledge from contexts or commonsense knowledge when doing MRC task. Thus, it is of significance for models to be able to reason with knowledge, especially commonsense knowledge.

Various deep learning models have been proposed and shown pretty good performance on MRC tasks (Parikh et al., 2019; Zhu et al., 2018; Sun et al., 2018; Xu et al., 2017). Majority of these approaches utilize sequence relevant neural networks such as GRU (Cho et al., 2014), LSTM (Hochreiter and Schmidhuber, 1997) and Attention mechanism (Vaswani et al., 2017) to model the implicit relation among passages, questions and answers.

As pre-trained language models have shown miraculous performance on several NLP tasks, a large number of methods utilize this pre-trained language model to extract textual level features in MRC tasks. (Zhang et al., 2019; Ran et al., 2019) compute the contextual representation of passages, questions and options separately with BERT and match the representation in downstream networks. They achieved the best results on RACE dataset at their submission time.

Shared task 1 in COIN workshop is a two-choice question task with short narrations about

* Equal contribution.
† Corresponding email: michaelwzhu91@gmail.com; zhuwei972@pingan.com.cn.

Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing, pages 93–98
Hongkong, China, November 3, 2019.

everyday scenarios, which is an extended version of SemEval 2018 Task 11 (Ostermann et al., 2018). Shared task 2 uses the ReCoRD dataset (Zhang et al., 2018a), a machine reading comprehension dataset in news articles. It annotates named entities in the news articles and additionally provides some brief bullet points that summarize the news. It then asks for cloze-style answers, filling in a blank in a sentence related to the news article. Accomplishing these tasks requires both the capability of reading comprehension and commonsense knowledge inference.

Our system is based on XLNet (Yang et al., 2019), a generalized auto-regressive pretraining method which achieves state-of-the-art results on many NLP tasks. For task 1, We first pre-train the model on multiple-choice question dataset RACE (Lai et al., 2017) to gain certain reading comprehension abilities. Afterwards, we mine commonsense knowledge by fine-tuning grounded commonsense inference dataset SWAG (Zellers et al., 2018) on XLNet instead of introducing knowledge graph of general knowledge such as ConceptNet (Speer et al., 2017) or WordNet (Miller, 1995). For task 2, other than utilizing XLNet's representation power, we also experiment on enhancing the representation and regularizing predicted prior of named entities, by concatenating the pre-trained embedding of WordNet of contextual word embedding. We finally implement a series of post-processing strategies to improve the model prediction results. Our system achieves state-of-the-art performance on the both shared tasks' official test data, even though we only train on the train sets and only submit single models.

2 Model settings

In this section, we present the system designs we experimented for the two shared tasks.

2.1 Pretrained language model Fine-tuning

As shown in Devlin et al. (2018) and Yang et al. (2019), the usual way to employ pre-trained language models in representing multiple text input is concatenation of text inputs in certain orders. For this section, the denotations mainly follow Yang et al. (2019) since we mainly use XLNet as the text encoder. Since the notations for Bert will be quite similar, which will not be included in this work.

For task 1, the inputs are a context passage (denoted as P), two queries (denoted as Q_i, $i = 1, 2$), and two answer options for each query, $A_{i,j}$, for $j = 1, 2$. Following Yang et al. (2019)'s solution on the RACE dataset, we concatenate the inputs as follows:

$$Concat\ {}_{i=1}^{2}[P, [SEP], QA_i, [SEP], [CLS]], \tag{1}$$

where QA_i is the concatenation of the query-answer pairs:

$$QA_i = Concat\ {}_{j=1}^{2}[Q_i, A_{i,j}]. \tag{2}$$

As for Task 2, the inputs are a context passage (denoted as P), which in this case is a piece of a newspaper article, and a assertive sentence (denoted as S) part of which is masked out, thus they are concatenated as follows:

$$Concat\ [P, [SEP], S, [SEP], [CLS]]. \tag{3}$$

After the text inputs are concatenated accordingly, they will go through XLNet to get a contextual representation. The output layer for the two tasks are different. For task 1, a fully-connected layer is put on the $[CLS]$ token's representation two give out the likelihood of which answer option is the answer. For task 2, the answer is selected from the context passage, thus we have to predict the start position and end position. Thus two fully-connected layers are needed, where the first is to estimate the likelihood of being the start position for each token, and the second combines the encoded representation and the output of the first fully-connected layer and predicts the end position.

2.2 Multi-funetuning

Finetuning a pre-trained language model on a small target task dataset has shown significant performance gains, as is shown in Devlin et al. (2018) and Yang et al. (2019). However, directly fine-tuning is proven not to be the most effective way, since although pre-trained LMs are known to generalize well, overfitting problem is still inevitable. Thus, related corpus or similar datasets are often used, such as Wang et al. (2019) and Phang et al. (2018), to form a multi-stage fine-tuning procedure. For example, Wang et al. (2019) first fine-tune on the MultiNLI datset before training the CB, RTE, and BoolQ tasks. The intuition behind why this multi-stage fine-tuning strategy works is

Dataset	Options	Sentence A	Sentence B
RACE	4	passage	query+option
SWAG	4	query	option
Task 1	2	passage	query+option

Table 1: Structure of inputs for the two supplementary tasks and the target task dataset

Dataset	Train	Dev
RACE	87866	4887
SWAG	73546	20006
Task 1	14191	2020

Table 2: Basic statistics for the three datasets involved in solving task 1

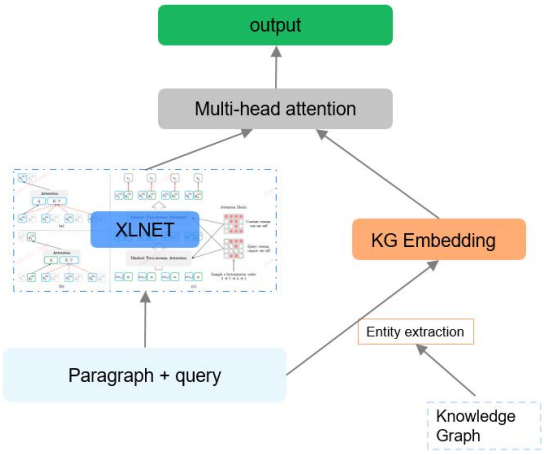

Figure 1: The architecture of our KG infusing model with XLNet as text encoder

that (a) to let the pre-trained LMs to adopt to the similar contextual environment, (b) and make the model more suitable for this specific task formation.

Due to the fact that task 1 dataset is small and during fine-tuning the original XLNet model overfits very quickly, we experimented on a multi-stage fine-tuning strategy. The first additional dataset we choose is RACE, which is relatively larger. Then we choose to fune-tune on SWAG, whose queries are similar to our target task and requires commonsense reasoning. Then we fine-tune till convergence on the task 1 train set.

Table 1 presents the structure of the inputs for the three datasets, where k is the number of options for each query. After each stage before the final fine-tuning, we disregard the final fully-connected output layer and use the updated XLNet layers to fine-tune on the next dataset.

2.3 Knowledge fusing

Besides the original basic fine-tuning architecture adopted by the XLNet, we also experiment on involving commonsense knowledge for context encoding, as is depicted in Figure 1.

The commonsense knowledge graph we use is the WordNet (Miller, 1995). The KG embedding is trained using DistMult (Yang et al., 2014a). First, we will match the phrases in the passage to entities in the WordNet, using Aho-Corasick algorithm (Arudchutha et al., 2014).[1] Then each token in the entity will be given the same embedding vector, which is the embedding of the entity in WordNet. Tokens not in any entity will be given a zero vector as embedding. The KG encoded text input will be incorporated with the encoded output of XLNet using a multi-head attention layer (Vaswani et al., 2017), where the XLNet encoded output acts as the query and the KG encoded output acts the key and value. Then the output layer is the same with answer span prediction layers described in the previous subsection.

[1] If a phrase is matched to multiple entities in the KG, we will take the average of all entity embeddings as the entity embedding for the phrase.

2.4 Answer Verification

To improve the prediction results of a model, we implement a series of answer verification strategies, which are the following:

- as there are additional entity information provided with the dataset, at the span predict stage, we filter invalid predicted spans according to whether it match a named entity
- if we can not find any entities in all predictions, we randomly select one from the entities provided to us
- some entity is a part of the '-' concatenation span, then we match the answer by its left or right concatenated contexts

3 Experiment

3.1 Dataset

Statistics for the datasets involved in training for task 1, which are RACE, SWAG and the official task 1 dataset are shown in Table 2. The statistics represent the total number of queries in the corresponding dataset. The final submission result on

the leader-board is calculated on official test data, which will not be published. Only training and development data for the task are available to us.

Task 2, which is the ReCoRD dataset (Zhang et al., 2018b) has 65, 000 queries on the train set and 10, 000 queries on the dev set. The answer for the ReCoRD dataset is not unique, since an entity is likely to be mentioned multiple times in a news article. Thus, we take each passage-query-answer-span as one sample during training, which can also be seen as a kind of data augmentation.

For both tasks, we only submit the models trained on the train sets of the target tasks.

3.2 Experimental setting

We use XLNet (large, cased) as the pre-trained language model. For task 1 dataset, we truncate the query-answer pair to a maximum length of 128, and set the maximum length of the passage-query-answer pair to 384. So the max length of the whole text inputs of one sample is 768. With a Tesla V100-PCIE-16GB GPU card, the batch size can only be set to be 2 on each card, thus we employ 8 GPUs for training. Firstly we fine-tune the original XLNet on RACE for 100,000 steps with the sequence length of 192, query-answer length of 96 and Adam optimizer leaning rate of 1e-6. Afterwards, we fine-tune the model on SWAG for 12,500 steps with the same parameters as RACE's. Eventually, the model is fine-tuned on the task 1 dataset till convergence, where the learning rate is set as 8e-6.

For task 2, the maximum length of the passage-querypair is set to be 384, in which the maximum length for the query is 64. During training the learning rate is 5e-6 and batch size is 4 on each GPU card.

When we try to infuse the KG into the XLNet, we use the OpenKE library (Han et al., 2018) to train the KG representions of WordNet. We choose DistMult (Yang et al., 2014b) as the embedding model, set the embedding size as 100, epoches as 10, batch size as 32 and the learning rate as 1e-4. The multi-head attention betwen the XLNet encoded output and the entity encoded output has the same number of attention head as the XLNet large model. During training, we will keep the KG embedding trainable. Besides multi-head attention, we also experiment using a whole transformer block, i.e., a multi-head attention layer followed by a position-wise feed-forward network, and combining the entity encoding by simply concatenating it onto the XLNet encoded output. For comparison, we switch XLNet with Bert (large model), and repeat the above experiments.

Model	Dev	Test
Human	-	97.4%
Final submission	91.44%	90.6%
XLNet	91.09%	-
XLNet+RACE	92.46%	-
XLNet+SWAG	89.36%	-
XLNet+RACE+SWAG	**92.76%**	-

Table 3: Main results on the task 1.

3.3 Results

The main experimental accuracy results for task 1 are shown in Table 3, in which human performance is provided by task organizers. Our system consists of fine-tuning XLNet on RACE and SWAG. We also conduct an ablation experiment to investigate the effects of the two external dataset. As a result, Table 3 illustrates that pre-training on RACE plays a significant role in the system. Origin XLNet achieves an accuracy of 91.09%, indicating its powerful text representation ability, especially in the reading comprehension task. Pre-training on SWAG without RACE does not improve the accuracy perhaps because SWAG misleads the model to better adjust its task formation, thus making it worse on the machine reading comprehension. Meanwhile, combining SWAG together with RACE makes sense, indicating the model can improve its commonsense inference ability.

We achieved the best performance on the official dev dataset with the training steps described in section 3.2 while our submission result on the official leader-board was obtained with fewer training steps due to queue submission time impact on Codalab. Despite being not fully trained, our system still achieve the best result with the accuracy of 90.6%, outperforming other participating teams.

For task 2, the results are presented on Table 4, where the bolded models are our submissions on the test leaderboard. Due to limited resources and, the results are not run multiple times, thus the results may be affected by random effects. We find out the original XLNet performs the best, significantly outperforming the Bert models. While it seems adding a commonsense KG is beneficial for

Model	Dev EM	Dev F1	Test EM	Test F1
Human	-	-	91.31 %	91.69 %
Bert	69.83 %	71.05 %	-	-
Bert + KG (multi-head attn)	71.08 %	72.69 %	-	-
Bert + KG (transformer)	70.36 %	71.84 %	-	-
Bert + KG (concat)	70.74 %	72.09 %	-	-
XLNet	80.64 %	82.10 %	81.46 %	82.66 %
XLNet + KG (multi-head attn)	80.31 %	81.62 %	-	-
XLNet + KG (transformer)	80.16 %	81.55 %	-	-
XLNet + KG (concat)	80.25 %	81.67 %	-	-
XLNet + answer verification	**82.72 %**	**83.38 %**	**83.09 %**	**83.74 %**

Table 4: The main results on task 2.

Bert, it does not help improving XLNet models. For the models with KG, regardless of what the underlying pre-trained langugae model is, multi-head attention works best on infusing the knowledge, and simple concatenation works better than adding a whole transformer block.

For task 2, implementing the answer verification process after we obtain the predictions of XLNet model boost the performance significantly, both on the dev set and the test set. Since we did not see significant improvements by adding KG into the model, we did not submit results from KG infused models.

Conclusions

To conclude, we have shown that XLNet, a recently proposed pre-trained language model, is powerful in text representation for machine reading tasks. Simply fine-tuning XLNet on the shared tasks already outperforms the other models which use Bert as text encoder. However, we demonstrate on task 1 that multi-stage fine-tuning on similar tasks can help providing more stable convergence and improve the final results significantly. For task 2, we also show that the model predictions can be improved by adding human designed post-processing strategies. We also experiments on incorporating commonsense KG into the architecture of XLNet, however, due to our limited experiments, we haven't obtain significant improvements by adding KG into the model, especially models based on XLNet. However, it is a direction worth further research.

References

S. Arudchutha, T. Nishanthy, and R. G. Ragel. 2014. String matching with multicore cpus: Performing better with the aho-corasick algorithm. In *IEEE International Conference on Industrial Information Systems*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

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*.

Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of EMNLP*, pages 139–144.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Divyansh Kaushik and Zachary C Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset.

Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. 2018. Semeval-2018 task 11: Machine comprehension using commonsense knowledge. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 747–757.

Soham Parikh, Ananya B Sai, Preksha Nema, and Mitesh M Khapra. 2019. Eliminet: A model for eliminating options for reading comprehension with multiple choice questions. *arXiv preprint arXiv:1904.02651*.

Jason Phang, Thibault Fvry, and Samuel R. Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Qiu Ran, Peng Li, Weiwei Hu, and Jie Zhou. 2019. Option comparison network for multiple-choice reading comprehension. *arXiv preprint arXiv:1903.03033*.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2018. Improving machine reading comprehension with general reading strategies. *arXiv preprint arXiv:1810.13441*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems.

Yichong Xu, Jingjing Liu, Jianfeng Gao, Yelong Shen, and Xiaodong Liu. 2017. Dynamic fusion networks for machine reading comprehension. *arXiv preprint arXiv:1711.04964*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014a. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *arXiv e-prints*, page arXiv:1412.6575.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014b. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *arXiv e-prints*, page arXiv:1412.6575.

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*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018a. Record: Bridging the gap between human and machine commonsense reading comprehension. *CoRR*, abs/1810.12885.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Benjamin Van Durme. 2018b. Record: Bridging the gap between human and machine commonsense reading comprehension.

Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019. Dual co-matching network for multi-choice reading comprehension. *arXiv preprint arXiv:1901.09381*.

Haichao Zhu, Furu Wei, Bing Qin, and Ting Liu. 2018. Hierarchical attention flow for multiple-choice reading comprehension. In *Thirty-Second AAAI Conference on Artificial Intelligence*.