Momentum Contrastive Pre-training for Question Answering

Minda Hu¹ Muzhi Li¹ Yasheng Wang² Irwin King¹
¹Dept. of Computer Science & Engineering, The Chinese University of Hong Kong
²Huawei Noah’s Ark Lab
{mindahu21, mzli, king}@cse.cuhk.edu.hk,
wangyasheng@huawei.com

Abstract

Existing pre-training methods for extractive Question Answering (QA) generate cloze-like queries different from natural questions in syntax structure, which could overfit pre-trained models to simple keyword matching. In order to address this problem, we propose a novel Momentum Contrastive Pre-training for Question Answering (MCROSS) method for extractive QA. Specifically, MCROSS introduces a momentum contrastive learning framework to align the answer probability between cloze-like and natural query-passage sample pairs. Hence, the pre-trained models can better transfer the knowledge learned in cloze-like samples to answering natural questions. Experimental results on three benchmarking QA datasets show that our method achieves noticeable improvement compared with all baselines in both supervised and zero-shot scenarios.

1 Introduction

The task of extractive Question Answering (QA), which aims to select an answer span from a passage given a query, is a major focus of NLP research. Currently, deep learning systems (Huang et al., 2018; Devlin et al., 2019; Wu et al., 2021) have achieved competitive results with humans on large-scale QA datasets (Rajpurkar et al., 2016, 2018; Yang et al., 2018; Kwiatkowski et al., 2019). However, the collection of high-quality natural QA pairs is still a labor-intensive task, especially for the construction of domain-specific QA systems. To alleviate such data availability restrictions, pre-training methods have been drawing increasing attention (Dhingra et al., 2018; Ram et al., 2021).

Typically, pre-training methods for extractive QA generate cloze-like query-passage pairs with text-matching techniques. For instance, the Span Selection Pre-Training (SSPT) method (Glass et al., 2020a) generates these pairs with the Wikipedia corpus using BM25. Nevertheless, these methods have two major issues. Firstly, the format of cloze-like queries differs much from natural queries asked by humans (see Fig. 1). In addition, models trained by cloze-like queries overfit in capturing the lexical overlaps between queries and passages, which restricts the capability of high-level semantic reasoning (Li et al., 2020; Hu et al., 2021). Since answering natural questions requires a comprehensive understanding of queries and passages, models pre-trained with cloze-like queries are not well aligned with downstream QA tasks.

To solve these issues, we utilize Contrastive Learning (CL) techniques to align knowledge learned in cloze-like samples to answering natural language questions and circumvent overfitting. Specifically, CL aims to learn representations by contrasting augmentations of different input data, which has been successfully applied in various research fields such as bioinformatics, graph learning, computer vision, and NLP (Han et al., 2022; Zhang et al., 2022; He et al., 2020; Yang et al., 2021; Pan et al., 2021). Nevertheless, existing CL methods for QA focus on query-passage matching (Yang et al., 2021; Caciularu et al., 2022) and multilingual embedding alignment (Pan et al., 2021), which have not been tailored to deal with issues of query format inconsistency and overfitting in matching lexical overlaps.

In this paper, we propose a Momentum Contrastive Pre-training for Question Answering (MCROSS) method.
ing (MCROSS) method for extractive QA. Specifically, MCROSS employs a momentum contrastive learning strategy along with the conventional answer prediction task to maximize the consistency of predicted answer distributions between cloze-like and natural query pairs and thus improves the performance of pre-trained models in answering natural language questions. We show the efficacy of our approach on standard English benchmark datasets. On SQuADv1.1 (Rajpurkar et al., 2016), MCROSS achieves about 2.7/3.5 percentage points gain on F1/Exact Match (EM) accuracy over BERT, and 1.4/1.8 percentage points improvement on the same metrics compared to SSPT (Glass et al., 2020a). MCROSS also consistently outperforms the baseline methods by a large margin on TriviaQA (Joshi et al., 2017) and NewsQA (Trischler et al., 2017) in supervised and zero-shot scenarios.

2 Method

The structure of our proposed MCROSS method is shown in Fig. 2. Along with the cloze-like samples \( s^{\text{cloze}} = (q_c, p, a) \) generated by SSPT (Glass et al., 2020a), we create positive natural samples \( s^{\text{natural}^+} = (q_n, p, a) \) containing natural queries \( q_n \) given passages and answers \( (p, a) \) from \( s^{\text{cloze}} \) with T5 answer-aware question generator (Raffel et al., 2020) which is fine-tuned with SQuADv1.1 training set. The details of pre-training datasets are illustrated in Appendix A.1. Though different in format, the cloze-like queries \( q_c \) and positive natural queries \( q_n \) are semantically similar since they have the same answers \( a \) in given supporting passages. Therefore, we expect that the predicted probability distribution of answer span in positive pairs \( (s^{\text{cloze}}, s^{\text{natural}^+}) \) should be closer. On the other hand, we generate negative query pairs \( (s^{\text{cloze}}, s^{\text{natural}^-}) \) with different answers or supporting passages. Correspondingly, the predicted answer span distribution of \( s^{\text{natural}^-} \) is expected to be far from that of \( s^{\text{cloze}} \). MCROSS achieves the goal with two pre-training tasks as follows.

2.1 Pre-Training Tasks

**Answer Term Prediction Task.** Given a training sample \( s = (q, p, a) \), the answer term prediction task aims to predict the token span \( a = (s_t, e_n) \) in the query \( q \) from the passage \( p \), where \( s_t \) and \( e_n \) are the start and end positions of the answer.

![Figure 2: An overview of the proposed MCROSS method.](image)

First, we prepare the concatenated token sequence \( T = [CLS] q [SEP] p [SEP] \) and leverage BERT (Devlin et al., 2019) to encode the context embeddings \( H \in \mathbb{R}^{(\text{len}(q)+\text{len}(p)+3) \times d} \) for each query-passage pair, where \( d \) is the dimension of embeddings. Then, the probability distribution \( Z = (z^{st_a}, z^{en_a}) \) of the start/end indices of answer span can be predicted as:

\[
    z^{st_a} = p(i = s_t) = \text{softmax}(L_{\text{start}}(H_i)),
    \]

\[
    z^{en_a} = p(i = e_n) = \text{softmax}(L_{\text{end}}(H_i)),
    \tag{1}
\]

where \( L_{\text{start}} \) and \( L_{\text{end}} \) are linear answer span prediction layers. The loss function of the answer term prediction task is defined below:

\[
    L_{\text{Answer}}(Z) = -\frac{1}{2} \left( \sum_{1 \leq i \leq N} \mathbb{1}(i = s_t) \log z^{st_a} + \sum_{1 \leq i \leq N} \mathbb{1}(i = e_n) \log z^{en_a} \right),
    \tag{2}
\]

where \( \mathbb{1}(\text{condition}) \) is the indicator function that returns 1 if the condition is satisfied and 0 otherwise. \( N \) is the length of the sequence \( T \).

**Contrastive Learning Task.** This task utilizes a contrastive loss function to guide the answer span distributions \( (Z^{\text{cloze}}, Z^{\text{natural}^+}) \) of positive sample pairs \( (s^{\text{cloze}}, s^{\text{natural}^+}) \) with same \( (p, a) \) to be closer, while keeping \( Z^{\text{cloze}} \) to be dissimilar from \( Z^{\text{natural}^-} \) of \( s^{\text{natural}^-} \) with different passages \( p^- \) or answers \( a^- \).

Inspired by the **Momentum Constrictive learning** (MoCo) (He et al., 2020), we maintain a large pool of consistent negative samples...
from several previous batch iterations to preserve more information on negative samples and obtain better pre-training objectives. Following MoCo, we employ a dual-encoder architecture (BERT\textsubscript{fast}, BERT\textsubscript{slow}) and maintain FIFO queues \((Q_{\text{slow}}^n, Q_{\text{ena}}^n)\) containing a large set of answer span distributions \((z_{\text{slow}}^-, z_{\text{ena}}^-)\) of negative samples predicted by BERT\textsubscript{slow}. To maintain queue consistency, for each batch iteration, parameters \(\theta_{\text{slow}}\) of BERT\textsubscript{slow} are only updated by the exponential moving average of \(\theta_{\text{fast}}\) of BERT\textsubscript{fast}. Here we have
\[
\theta_{\text{slow}} = m\theta_{\text{slow}} + (1 - m)\theta_{\text{fast}},
\]
where \(m\) is the momentum coefficient. We regulate the output of the answer-span classification layer with InfoNCE loss to bring positive pairs \((s, s^+\)) closer to each other and push negative ones apart:
\[
\mathcal{L}_{\text{MoCo}}(z_{\text{fast}}, z_{\text{slow}^+}, Q) = \log \frac{\exp(D(z_{\text{fast}}, z_{\text{slow}^+})/\tau)}{\sum_{z_{\text{slow}^-} \in Q} \exp(D(z_{\text{fast}}, z_{\text{slow}^-})/\tau)},
\]
where \(z_{\text{fast}}\) and \(z_{\text{slow}^+}\) represent the start or end distributions of positive pairs, \(\tau\) is the softmax temperature, and \(Q\) denotes queues containing distributions \(z_{\text{slow}^-}\) of negative samples \(s^-\) which have \((p^-, a^-)\) being different from \(s^+\). \(D\) is the similarity function measuring distances between answer span distributions. Cosine similarity is utilized in the original MoCo method (He et al., 2020). However, we argue that KL-Divergence is more suitable for the measurement of differences between two probability distributions, which is evaluated and validated in Appendix C.

### 2.2 Variants of the MCROSS Method

There are two variants of the MCROSS method with different loss functions, the first is involved with the unilateral loss function, and the second one with the bilateral loss.

**MCROSS (UNI).** In addition to the original SSPT training loss \(\mathcal{L}_{\text{Answer}}(Z^c)\), we add \(\mathcal{L}_{\text{MoCo}}\) to maximize the consistency of prediction between \(s_{\text{cloze}}\) and \(s_{\text{natural}}\) with little overhead in the unilateral version of MCROSS. Here, we have the unilateral MoCo loss
\[
\mathcal{L}_{\text{UNI}}(s_{\text{cloze}}, s_{\text{natural}}) = \mathcal{L}_{\text{Answer}}(Z^c) + \lambda_{\text{MoCo}}/2 \cdot [\mathcal{L}_{\text{MoCo}}(z_{\text{ena}^-}, Q_{\text{ena}^n}, z_{\text{ena}^-}, Q_{\text{ena}}^n)] + \mathcal{L}_{\text{MoCo}}(z_{\text{ena}^-}, z_{\text{ena}^-}, Q_{\text{ena}^n}),
\]

### 3 Experiments

We use the following three English span-extraction QA datasets to evaluate pre-trained models on F1/EM metrics in both supervised and zero-shot scenarios. Specifically, TriviaQA (Joshi et al., 2017) and NewsQA (Trischler et al., 2017) are out-of-domain datasets from MRQA 2019 Shared

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
<th>EM</th>
<th>0-shot F1</th>
<th>0-shot EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>87.43</td>
<td>79.50</td>
<td>8.00</td>
<td>0.06</td>
</tr>
<tr>
<td>SSPT</td>
<td>88.75</td>
<td>81.25</td>
<td>22.88</td>
<td>15.42</td>
</tr>
<tr>
<td>SSPT\textsuperscript{l}</td>
<td>89.25</td>
<td>82.13</td>
<td>64.41</td>
<td>44.44</td>
</tr>
<tr>
<td>MCROSS(UNI)</td>
<td>89.95</td>
<td>81.60</td>
<td>24.78</td>
<td>16.58</td>
</tr>
<tr>
<td>MCROSS(BI)</td>
<td>90.11</td>
<td>83.03</td>
<td>65.68</td>
<td>45.40</td>
</tr>
<tr>
<td>w/o (\mathcal{L}_{\text{MoCo}})</td>
<td>89.92</td>
<td>82.96</td>
<td>65.20</td>
<td>44.82</td>
</tr>
</tbody>
</table>

Table 1: Results on SQuADv1.1 dataset. (in domain)
Task (Fisch et al., 2019). Since SQuADv1.1 (Rajpurkar et al., 2016) dataset is utilized to fine-tune the T5 question generator, it is also included to examine the in-domain QA performance.

The experimental settings, implementation details, and complexity analysis are presented in Appendix A, B, and D.

3.1 Baselines
We compare the MCROSS with the following four baselines:
• BERT: The 12-layer BERT model released by (Devlin et al., 2019).
• MRQA-BERT: The official multi-task baseline BERT-base model from MRQA2.
• SSPT: The span selection pre-training method proposed by (Glass et al., 2020b). This method trains models with cloze-like samples $s_{\text{cloze}}$ using answer term prediction loss $L_{\text{Answer}}(Z^n)$ in Eq. (7).
• SSPT†: SSPT method trained with only natural samples $s_{\text{natural}}$ using answer term prediction loss $L_{\text{Answer}}(Z^n)$ in Eq. (7).

3.2 Experiment Results
**SQuADv1.1.** Table 1 shows the performance of all models on SQuADv1.1. Compared with the BERT baseline without extended pre-training, three variants of the proposed MCROSS method increase the F1/EM metrics by at least 1.5/2.1 percentage points. In addition, MCROSS(w/o $L_{\text{MoCo}}$) achieves discernible boosts among all metrics compared to baseline SSPT. It proves that the combination of cloze-like and natural samples in the QA task can endow the pre-trained model with a better understanding on supporting passages. Moreover, the gap between MCROSS(BI) and MCROSS(UNI) indicates that models trained only with cloze-like samples will be insensitive to natural questions. It should be also noticed that there exists a performance gap between SSPT† and MCROSS(UNI) among all metrics. This is attributed to the fact that the natural samples used in SSPT† are generated by the T5 question generator (Raffel et al., 2020). Since the T5 generator is fine-tuned on the training set of SQuADv1.1, it is well fitted in the domain of SQuAD. Compared to MCROSS(UNI) using cloze-like samples, SSPT† is trained with the natural samples containing more domain knowledge of the dataset, thus surpassing MCROSS(UNI).

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
<th>EM</th>
<th>0-shot F1</th>
<th>0-shot EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>64.28</td>
<td>49.17</td>
<td>3.55</td>
<td>0.02</td>
</tr>
<tr>
<td>SSPT</td>
<td>65.86</td>
<td>50.62</td>
<td>9.58</td>
<td>6.05</td>
</tr>
<tr>
<td>SSPT†</td>
<td>65.60</td>
<td>50.52</td>
<td>34.03</td>
<td>15.67</td>
</tr>
<tr>
<td>MRQA-BERT</td>
<td>66.80</td>
<td>50.80</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MCROSS(UNI)</td>
<td>66.45</td>
<td>51.33</td>
<td>10.90</td>
<td>6.55</td>
</tr>
<tr>
<td>MCROSS(BI)</td>
<td>67.90</td>
<td>52.18</td>
<td>34.82</td>
<td>14.98</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
<th>EM</th>
<th>0-shot F1</th>
<th>0-shot EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>62.86</td>
<td>57.39</td>
<td>3.25</td>
<td>0.04</td>
</tr>
<tr>
<td>SSPT</td>
<td>70.93</td>
<td>65.27</td>
<td>26.89</td>
<td>22.47</td>
</tr>
<tr>
<td>SSPT†</td>
<td>69.09</td>
<td>63.24</td>
<td>42.01</td>
<td>33.99</td>
</tr>
<tr>
<td>MRQA-BERT</td>
<td>71.60</td>
<td>65.60</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MCROSS(UNI)</td>
<td>72.06</td>
<td>66.41</td>
<td>28.28</td>
<td>23.48</td>
</tr>
<tr>
<td>MCROSS(BI)</td>
<td>73.77</td>
<td>68.05</td>
<td>47.79</td>
<td>39.53</td>
</tr>
<tr>
<td>w/o $L_{\text{MoCo}}$</td>
<td>73.30</td>
<td>67.72</td>
<td>46.63</td>
<td>38.54</td>
</tr>
</tbody>
</table>

**NewsQA and TriviaQA.** Table 2 and Table 3 show the results on NewsQA and TriviaQA dataset from MRQA 2019 Shared Task. It is noticeable that MCROSS(BI) performs the best among all methods on F1 metrics. Compared with the state-of-the-art baseline SSPT, it achieves an improvement of 2.0 F1 score and 1.6 EM accuracy on NewsQA. On the TriviaQA dataset, MCROSS(BI) also surpasses SSPT by 2.8 percentage points on F1 and EM accuracy. Furthermore, the improvement on TriviaQA of MCROSS(BI) over MCROSS(w/o $L_{\text{MoCo}}$) demonstrates the effectiveness of $L_{\text{MoCo}}$ in the out-of-domain setting. In both datasets, the great improvement on zero-shot F1/EM of SSPT† over SSPT exhibits that SSPT† has gained the domain knowledge in understanding natural questions from SQuAD. In comparison with SSPT†, MCROSS(BI) significantly boosts zero-shot QA performance by 5.6/6.2 percentage points of F1/EM accuracy.

In contrast to the SQuAD dataset with in-domain settings, MCROSS(UNI) has gained a noticeable advantage over SSPT† on both NewsQA and TriviaQA datasets, hinting that SSPT† is overfitted to the natural samples of SQuAD with extra domain knowledge. Interestingly, MCROSS(BI) performs worse in zero-shot scenarios on NewsQA dataset than MCROSS(w/o $L_{\text{MoCo}}$). This may be due to the fact that NewsQA has 32.7% samples that can be easily answered by simple text-matching (Trischler et al., 2017), into which MCROSS(w/o $L_{\text{MoCo}}$) is overfitted to capture lexical overlaps.

---

4 Conclusion

This paper presents a novel pre-training method MCROSS for extractive QA which contains two tasks: 1) contrastive learning and 2) answer term prediction. Specifically, MCROSS adapts MoCo frameworks to maintain consistency in answering cloze-like and natural questions, enabling pre-trained models to have a more comprehensive understanding of supporting passages. The empirical experiments on three public datasets demonstrate that our approach can obtain noticeable improvements in extractive QA tasks in supervised and zero-shot scenarios.

5 Limitations

Although MCROSS can already obtain satisfactory QA performance, due to limited time and computational resources, we only use 5 million cloze-like samples for pre-training, which is one-twentieth of the scale of original SSPT experiments.

References


A Experimental Settings

A.1 Pre-Training Dataset

Cloze-like Samples. We follow the method proposed by SSPT (Glass et al., 2020b) to generate cloze-like pre-training instances. Specifically, we first choose a sentence in the Wikipedia corpus and randomly replace an entity or noun phrase with the special token [BLANK]. The new sentence will be considered as a query. Based on the query, we retrieve a paragraph that contains the masked term as its corresponding passage. Due to the time and computational resource limitations, we finally collect 5 million unique pre-training examples in English for our pre-training, which is one-sixth of the original SSPT method.

Natural Samples. For each $s_{\text{cloze}} = (q_c, p, a)$, we use a public T5-base model$^3$ to generate natural queries $q_n$ given $(p, a)$. It is fine-tuned for answer-aware question generation using the SQuADv1.1 train set. If answer $a$ is empty, $q_n$ is not generated and loss functions related to $s_{\text{natural}}$ will not be included during training. We do not apply extra filtering or other quality control operations for generated questions.

A.2 Evaluation

We use the official model evaluation scripts of SQuADv1.1$^4$ and MRQA$^5$ to calculate the F1 and EM metrics between predictions and ground truth answers. Besides, we employ zero-shot F1 and zero-shot EM to quantify the performance of models without fine-tuning. Official training and development sets are used in all experiments.

B Implementation Details

All model architectures are based on the 12-layer BERT-base model with 110M parameters, and we employed Google's original implementation of the BERT model published in the Huggingface$^6$ to build up our encoder. The training batch size for pre-training and fine-tuning are 32 and 8. Total training steps for MCROSS and other baseline models are both kept to 156,250. The max sequence...
Table 4: Comparison between choosing $D_{KL}$ and $D_{Cosine}$

<table>
<thead>
<tr>
<th>Dataset</th>
<th>F1</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQuADv1.1</td>
<td>88.95</td>
<td>81.60</td>
</tr>
<tr>
<td>NewsQA</td>
<td>66.45</td>
<td>51.33</td>
</tr>
<tr>
<td>TriviaQA</td>
<td>72.06</td>
<td>66.41</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>F1</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQuADv1.1</td>
<td>88.47</td>
<td>80.79</td>
</tr>
<tr>
<td>NewsQA</td>
<td>65.98</td>
<td>50.40</td>
</tr>
<tr>
<td>TriviaQA</td>
<td>69.73</td>
<td>64.24</td>
</tr>
</tbody>
</table>

length of the transformer encoder is 384. In all experiments, we use the Adam optimizer with a learning rate of 2e-5. The MoCo momentum $m$ is set to 0.999, queue size 32,000, moco ratio $\lambda_{MoCo}$ 1.0 and temperature $\tau$ 0.05. Models are implemented on PyTorch. When the passage length exceeds the max sequence length, we follow the sliding window strategy proposed for BERT transformers in both fast/slow encoder structures.

During QA prediction, we apply the constraint $st_a \leq en_a$ to filtering out implausible answer spans and then select the ones with the highest joint probability $p(st_a) \times p(en_a)$ as results.

C Design Choice: KL Divergence

The performance gain of the MCROSS(UNI) method choosing KL divergence $D_{KL}$ over cosine similarity $D_{Cosine}$ as similarity function $D$ is shown in Table 4.

D Space and Time Complexity

The additional parameters of the answer prediction layer come from Eq. (1). The number of parameters is $(768 + 1) \times 2 = 1,538$, which is negligible in front of the parameters in BERT-base (110M). Training of the SSPT baselines and the MCROSS method took approximately 40 hours each on 8 V-100 GPUs.