From Good to Best: Two-Stage Training for Cross-Lingual Machine Reading Comprehension

Nuo Chen[1*], **Linjun Shou**[2†], **Ming Gong**[2], **Jian Pei**[3]
[1]ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2] STCA NLP Group, Microsoft, Beijing
[3]School of Computing Science, Simon Fraser University
nuochen@pku.edu.cn, {lisho,migon}@microsoft.com, jpei@cs.sfu.ca

Abstract

Cross-lingual Machine Reading Comprehension (xMRC) is challenging due to the lack of training data in low-resource languages. The recent approaches use training data only in a resource-rich language like English to fine-tune large-scale cross-lingual pre-trained language models. Due to the big difference between languages, a model fine-tuned only by a source language may not perform well for target languages. Interestingly, we observe that while the top-1 results predicted by the previous approaches may often fail to hit the ground-truth answers, the correct answers are often contained in the top-k predicted results. Based on this observation, we develop a two-stage approach to enhance the model performance. The first stage targets at recall: we design a hard-learning (HL) algorithm to maximize the likelihood that the top-k predictions contain the accurate answer. The second stage focuses on precision: an answer-aware contrastive learning (AA-CL) mechanism is developed to learn the fine difference between the accurate answer and other candidates. Our extensive experiments show that our model significantly outperforms a series of strong baselines on two cross-lingual MRC benchmark datasets.

Introduction

Machine Reading Comprehension (MRC) has been intensively studied in the Natural Language Understanding community in the past years (Rajpurkar et al. 2016; Yu et al. 2018; Chen et al. 2021; Seo et al. 2017; Liang et al. 2021; Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018; Reddy, Chen, and Manning 2019; You et al. 2020). When scaling out MRC to multiple languages, i.e., the task of cross-lingual MRC or xMRC for short, one challenge is the lack of training data in low-resource languages, where no training examples are available. To tackle this challenge, recent approaches build on large-scale cross-lingual pre-trained language models, such as mBERT (Pires, Schlinger, and Garrette 2019) and XLM-R (Conneau et al. 2019). These pre-trained models map the representations of different languages into a universal semantic space, where the expressions in different languages are represented close to each if they have similar meanings. The cross-lingual pre-trained models are then fine-tuned by training data only in a source language (e.g., English), and finally applied to various target languages. This approach has shown promising results on tasks such as entity recognition (Liang et al. 2021; Kruengkrai et al. 2020), question answering (Asai et al. 2018; Zhou et al. 2021), as well as xMRC (Yuan et al. 2020; Liang et al. 2021). However, due to the big difference between languages, the model fine-tuned only on the source language may not perform well on target languages.

Language	en	es	ar	hi	vi
EM score	64.24	48.30	35.14	41.93	42.36

Table 1: EM scores across different languages on MLQA dataset.

Table 1 is the results from our empirical study by applying the previous approach.specificlly, we use English data to fine-tune the cross-lingual XLM-R model (Conneau et al. 2019) on MLQA dataset (Lewis et al. 2020). The numbers in the table are exact match (EM) scores, which is a widely-adopted metric for MRC task to evaluate the match degree between the model predicted result and the ground-truth answer. For each case, we take the top-1 output from the model as the predicted result. From Table 1, we clearly see the result on English is much better than that on other languages. The reason is that the model fine-tuned by only English training data. Meanwhile, the model can still achieve 35 to 48 EM scores on non-English languages even though it has never been trained by any examples from those languages. This suggests that the model has inherited certain extent of language transfer capability from the cross-lingual PLMs.

We then extend the set of model predicted results by including top-k outputs from the model. That is, we regard the model "successfully" predicts the answer if any one in the top-k outputs matches the ground-truth answer. The modified EM scores with varying numbers of k are illustrated in Table 2 (note the numbers in the column of "Top-1" are just those in Table 1). From Table 2, we can see the scores have substantial gains for all languages when we increase k. The gain in English is the smallest (around 10 points when k=10 vs. k=1), since the model has been well fine-tuned on this language by the native training examples. In other lan-

[*]Work done during internship at Microsoft STCA.
[†]Corresponding Author

Language	Top-1	Top-3	Top-5	Top-10
en	64.24	73.06	75.69	75.76
es	48.30	60.32	66.04	71.18
de	46.43	60.99	67.13	72.17
ar	35.14	48.24	52.33	57.13
hi	41.93	57.70	63.32	70.15
vi	42.36	58.25	61.98	66.06

Table 2: EM scores among different top-k answer predictions on MLQA dataset, respectively.

guages, the gains are much larger, i.e., more than 20 points when k=10 vs. k=1. This observation discloses huge potential in the top-k results. Intuitively, it suggests that the model has been empowered with the ability to roughly distinguish good results with bad ones. However, without sufficient training examples, it is not powerful enough to rank the most accurate result at the top-1 position.

The analysis of results in Table 2 motivates us to decompose the training for xMRC model into two stages. The first stage targets at recall at top-k, which maximizes the likelihood for the accurate answer to be included in the set of top-k results. For this purpose, we design a hard-learning (HL) algorithm to learn the margin between good answers and bad ones. The second stage focuses on precision at top-1. We propose an answer-aware contrastive learning (AA-CL) mechanism to enable the model to further distinguish the accurate answer from the other candidates. Instead of selecting random or in-batch negatives, AA-CL constructs hard-negatives using the candidate that is most similar with (but not equal to) the ground-truth answer in the top-k prediction set at each update. Such hard-negatives help the model improve precision at top-1.

Our technical contributions are summarized as follows:

- We conduct an in-depth study on the xMRC task, make interesting observations, and design a novel two-stage approach based on the observations.
- We carry out extensive experiments and verify that our approach significantly surpasses previous state-of-the-art cross-lingual PLMs on two popular benchmarks.

The rest of the paper is organized as follows: We first review related work in Section 2, and then describe our proposed method detaily in Section 3. We report the extensive experiment results in Section 4, and further conduct ablation studies and analysis in Section 5. Finally, we conclude the paper in Section 6.

Related Work

Cross-Lingual Machine Reading Comprehension Recently, a considerable amount of literature has been published on cross-lingual machine reading comprehension (xMRC). A naive but efficient way is based on the machine translation system, which translates the training data in a rich-resource languages into other low-resource target language. Given the translated data, Cui et al. (2019) proposed to use back-translation for xMRC. Singh et al. (2019) developed a new translation-based data augmentation method for multilingual training. Unfortunately, all these methods heavily rely on the high-quality translation systems. On the other line, a school of approaches (Huang et al. 2019; Liang et al. 2020; Conneau et al. 2019) based on large-scale multilingual pre-trained language models (PLMs) have been proposed. And a series of experiments prove that these PLMs can achieve superior performances even if in zero-shot or few-shot setting. More recently, several efforts have been made to further improve the PLMs performance in xMRC. To address the answer boundary problem in low-resource languages, Yuan et al. (2020) proposed several auxiliary tasks on top of PLMs so as to improve the model performance. Following the line, Liang et al. (2021) presented a calibration neural network in a pre-training manner. Nevertheless, none of these studies explore to utilize top-k predictions from a base model as weak supervisions to train more robust models for xMRC.

Contrastive Learning Nowadays, Contrastive learning (Hadsell, Chopra, and LeCun 2006) has been seen as a promising way to build on learning effective representations by pulling together semantically close neighbors (*positive*) in a shared embedding space, and pushing apart non-neighbors (*negatives*). Contrastive learning objective has been particularly successful in different contexts of deep learning(Gao, Yao, and Chen 2021). Wu et al. (2020) proposed several sentence-level augmentation strategies to obtain a noise-invariant representation for down-stream tasks, such as text similarity and sentiment classification. Most recently, Gao, Yao, and Chen (2021) developed a simple contrasive learning method via using dropout (Srivastava et al. 2014) as noise. Concretely, they passed the same sentence into the PLMs twice and obtained positive pairs by applying dropout masks randomly. Although contrastive learning achieves significantly success in various natural language processing tasks, the context of question answering is less explored by research communities, especially for MRC. In this paper, we focus on a more challenge scenario: we propose AA-CL to leverage hard-negatives from highly confident predictions for xMRC, in which the hard-negatives are consistently updated during training.

Model

In this section, we aim to describe our proposed methods (See Figure 1) in detail. First, we introduce the problem formulation of xMRC. Then we describe the baseline model of our work. Last, we illustrate the hard-learning (HL) algorithm and answer-aware contrastive learning (AA-CL) sequentially.

Problem Formulation

The problem of xMRC studied in this paper can be formulated as follows. In this work, assume that our labeled data collection $\mathcal{D}_s \in \{q_i, p_i, a_i\}_i^N$ in a source language (rich-resource). Specifically, $\{q_i, p_i, a_i\}$ denotes the i-th triplets of $\{question, passage, answer\}$ in the training data. And we focus on the span-extraction MRC setting, where each answer $a_i = (a_{i,s}, a_{i,e})$ is a segment of text that appears in p_i, where $a_{i,s}, a_{i,e}$ denote the start and end positions of the

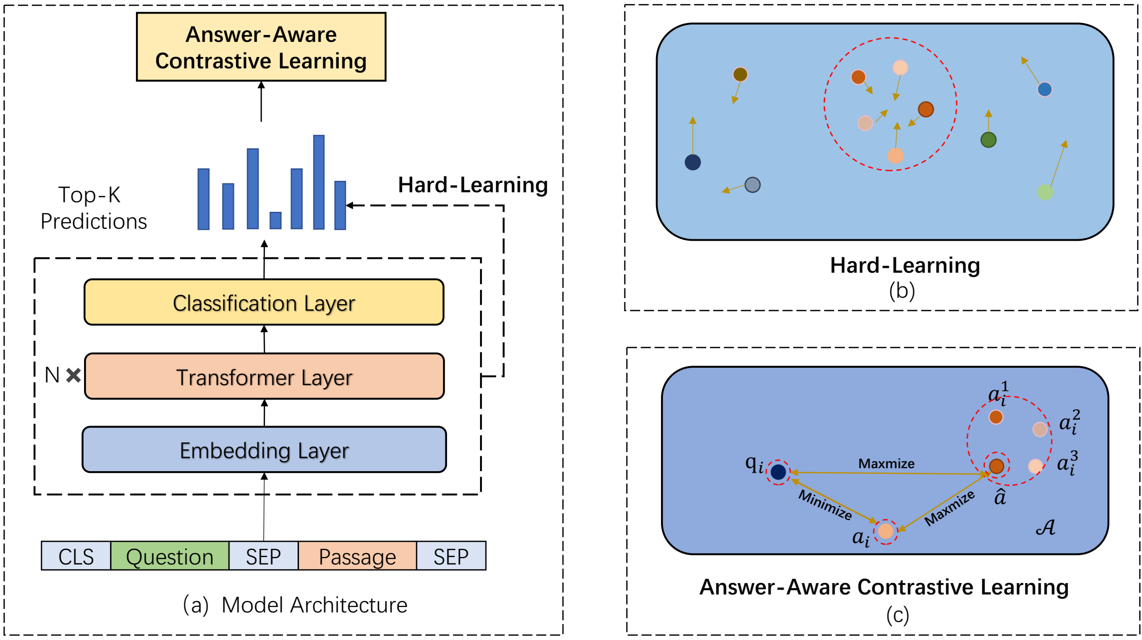

Figure 1: Overview of our proposed method.

ground-truth answer. The goal is to train a powerful model $\mathcal{M}$ on $\mathcal{D}_s$, and $\mathcal{M}$ can be able of performing well in other low-resource target languages.

Base Model $\mathcal{M}$

Our model is built on top of the powerful cross-lingual PLMs such as multilingual BERT and XLM-Roberta. Thereafter, the input question q_i and p_i are concatenated with two special tokens `[SEP]` and `[CLS]` to form the input sequence $\mathbf{X}$, as shown in the Figure 1 (a). `[CLS]` is used to mark the begin of the input sentence and `[SEP]` is responsible for separating the passage and question. We then feed $\mathbf{X}$ into the encoder, and produce contextualized token representations $\mathcal{X} \in \mathbf{R}^{l \times d}$:

$$\mathcal{X} = \mathcal{H}(\mathbf{X}) \tag{1}$$

where $\mathcal{H}$ is last encoder layer of cross-lingual PLMs, l is the max length of input sequence and d is the vector dimension of each token, separately.

Then, to predict the start position and end position of the correct answer span in $\mathbf{X}$, the probability distributions are induced over the entire sequence by feeding $\mathcal{X}$ into a linear classification layer and followed by a softmax function.

$$\mathcal{P}(s = i|\mathbf{X}), \mathcal{P}(e = i|\mathbf{X}) = \mathtt{softmax}(\mathcal{W} \cdot \mathcal{X}^T + b) \tag{2}$$

where $\mathcal{W} \in \mathbf{R}^{2 \times d}$. In the typically supervised setting, we can train a model $\mathcal{M}$ by optimizing the following function given the input q_i and p_i:

$$\begin{aligned}\mathcal{L}_{mrc} &= -\mathtt{log}\mathcal{P}(s = a_{i,s}|\mathbf{X}) - \mathtt{log}\mathcal{P}(e = a_{i,e}|\mathbf{X}) \\ &= -\mathtt{log}\mathcal{P}(s = a_{i,s}|p_i, q_i) - \mathtt{log}\mathcal{P}(e = a_{i,e}|p_i, q_i)\end{aligned} \tag{3}$$

Although, this approach achieves superior performances in xMRC, it only considers the top-1 predicted result while optimizing the model with cross-entropy loss, ignoring many correct predictions exits in top-k confident predictions, and thus, making the model sub-optimized. We overcome this issue by (1) developing a hard-learning algorithm with utilizing a pre-obtained n-best prediction set, and (2) proposing an answer-aware contrastive learning mechanism to leverage hard-negatives over training. We illustrate these two strategies in the following sections, separately.

Hard-Learning Algorithm

In this component, we aim to develop a hard-Learning (HL) algorithm during fine-tuning to maximize the likelihood for the accurate answer to be included in the set of top k predicted results, which is from pre-obtained highly confident predictions of a basic model. That is, HL enables the model to focus on the spans which are similar with the ground-truth answer to achieve the goal of recall, as shown in Figure 1 (b).

Definition Inspired by (Min et al. 2019), we define the correct answer for each question as a particular derivation that a model is required to solve for the answer prediction. Given a question q_i and a passage p_i, Let $\mathcal{Z} = \{z_1, z_2, ..., z_{k-1}, a_i\}$ be the set which contains top-k possible predictions from a baseline model (i.e., XML-R). Seen from Table 3, we assume it contains an unique correct answer[1] (a_i) that the model wants to learn to find, and potentially other ones which are hard to classify.

[1] If the correct answer doesn't occur in top-k predictions, we will replace the last one in $\mathcal{Z}$ with the correct answer.

An example from MLQA training dataset.
Question: To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France? **Passage:** Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with... Virgin Mary reputedly appeared to **Saint Bernadette Soubirous** in 1858. At the end of the main drive... **Answer: Saint Bernadette Soubirous** $\mathcal{Z}$ **(top-k predictions) :** Saint Bernadette, Bernadette, **Saint Bernadette Soubirous**, Saint, Saint...

Table 3: Examples of the input, answer text and Z. The correct answer is in bold. And in this example, the correct answer occurs in top-3 predictions from the model.

Algorithm In our work, the model not only has access to q_i and p_i, but also $\mathcal{Z}$, and then we assume that each z_l in $\mathcal{Z}$ can be seen as the *true* solution for the given question. Intuitively, we compute the maximum marginal likelihood (MML) to marginalize the likelihood of each $z_l \in \mathcal{Z}$ given the q_i and p_i, and the model can be optimized by the following loss function:

$$\mathcal{L}_{mml} = -\texttt{log} \sum_{z_l \in \mathcal{Z}} \mathcal{P}(z_l | q_i, p_i) \tag{4}$$

However, directly using MML for optimization can cause the model to fit on the noisy span labels contained in $\mathcal{Z}$. Specifically, in our settings, elements in $\mathcal{Z}$ can be categorized into three types: (1) ground-truth answer span (only 1); (2) spans which only match the start position or end position of the correct one; (3) spans which mismatch both the start and end positions. For the latter two types, the model is supposed to give lower or even zero probability. For example, as presented in Table 3, top-1 prediction "Saint Bernadette" and top 4 prediction "Saint" match the start position of the correct one but mismatch the end position. But when minimizing MML, it can assign high probabilities to any element in $\mathcal{Z}$. To address this problem, we utilize a HL algorithm where different weights are assigned to each element from $\mathcal{Z}$. Then the model can be optimized via a re-written standard cross-entropy loss as:

$$\mathcal{L}_{Hard} = -\sum_{l=1}^{k} w_l \texttt{log} \sum_{z_l \in \mathcal{Z}} \mathcal{P}(z_l | q_i, p_i) \tag{5}$$

where w_l is the learnable parameter.

Answer-Aware Contrastive Learning

HL encourages the model to focus on the spans which are similar with the ground-truth answer. Then, we deploy an answer-aware contrastive learning (AA-CL) mechanism to target on precision at top-1, which constructs hard-negatives using the most likely prediction with the ground-truth answer at each update to learn noise-invariant representations, which is complementary with the proposed HL algorithm. Therefore, to correctly identify hard-negatives, its relationship with the positive (ground-truth answer) must be carefully reasoned by the model, as shown in Figure 1 (c). In particular, we get the top-k answer predictions $\mathcal{A} = \{a_i^1, a_i^2, ..., a_i^k\}$ in each back propagation. And we select the one of $\mathcal{A}$ as the hard-negative example, which has the largest similarity with the ground-truth answer a_i[2]. This can be seen as an effective coarse-to-fine negative selection strategy. Formally:

$$\mathrm{a}_i^{(l)} = \mathcal{F}(\mathcal{H}(a_i^{(l)})) \tag{6}$$

$$\hat{a} = \texttt{max}_{a_i^l \in \mathcal{A}} \Psi(\mathrm{a}_i^l, \mathrm{a}_i) \tag{7}$$

where $\Psi(,)$ denotes the cosine similarity function and $\mathcal{F}$ means the mean-pooling operation. In this work, we argue that the similarity between the input question and the ground-truth answer is higher than others in $\mathcal{A}$. Hence, we can get the positive question-answer pair (q_i,a_i) and hard-negative pair (q_i,$\hat{a}$). For each pair, we use the contrastive objective to establish their correspondence among them in a shared semantic latent space:

$$\mathcal{L}_{\text{contrast}} = -\log \frac{\exp(\Psi(\mathbf{r}_q, \mathbf{r}_{a(pos)})/\tau)}{\sum_{n=1}^{B} \exp(\Psi(\mathbf{r}_q, \mathbf{r}_{a(n)})/\tau)}, \tag{8}$$

where B and τ are mini-batch and temperature. $\mathbf{r}_q$ and $\mathbf{r}_{a(pos)}$ denote the representation of question q_i and a_i. By this means, unlike only selecting negatives randomly or in-batch negatives, we also introduce the hard-negatives from high confidence predictions of the model over training, and thus, $\mathcal{M}$ can obtain the coarse-to-fine presentations in token-level. During the fine-tuning, $\mathcal{M}$ is optimized via $\mathcal{L}_{\text{Hard}}$ and $\mathcal{L}_{\text{contrast}}$ with the weighted ratio:

$$\mathcal{L}_{\text{final}} = \alpha \mathcal{L}_{\text{contrast}} + (1-\alpha)\mathcal{L}_{\text{hard}} \tag{9}$$

Experiments

Datasets and Evaluation Metrics

We evaluate our proposed methods on two popular datasets, MLQA (Lewis et al. 2020) and XQUAD (Asai et al. 2018), to examine the effectiveness.

MLQA is a popular xMRC benchmark, which covers various languages. We evaluate our methods on six languages: including *English, Arabic, German, Spanish, Hindi, Vietnamese*.

XQUAD is another dataset for evaluating the cross-lingual model performances, which consists of 11 languages. Similar to the setting above, we test our method with the same six languages in our experiments under the zero-shot and translation train setting.

We use two evaluation metrics, Exact Match (EM) and Macro-averaged F1 score (F1), which are popularly used for

[2] We present more detailed analysis in the Appendix.

Setting	Models	en	es	de	ar	hi	vi	Avg.
zero-shot	m-BERT	77.70 / 65.30	64.30 / 46.60	57.90 / 44.30	45.70 / 29.80	43.80 / 29.70	57.10 / 38.60	57.80 / 42.40
	XLM	74.90 / 62.40	**68.00 / 49.80**	62.20 / 47.60	54.80 / 36.30	48.80 / 27.30	61.40 / 41.80	61.70 / 44.20
	XLM-R$_{base}$	77.86 / 64.24	66.18 / 48.30	60.82 / 46.43	55.20 / 35.14	59.93 / 41.93	64.89 / 42.36	64.14 / 46.00
	Info-XLM	79.15 / 64.80	67.07 / 48.49	58.24 / 46.00	55.15 / 38.78	59.66 / 43.98	64.44 / 43.28	64.25 / 47.60
	Ours	**79.03 / 65.59**	67.52 / 49.56	**62.98 / 48.70**	**57.68 / 39.40**	**61.79 / 44.70**	**66.74 / 45.34**	**66.00 / 48.88**
translate	XLM-R$_{base}$	77.15 / 64.41	68.51 / 50.40	62.24 / 47.99	56.60 / 38.42	61.39 / 43.93	66.70 / 44.36	65.45 / 48.25
	LAKM	80.00 / 66.80	65.90 / 48.00	60.50 / 45.50	-	-	-	-
	CalibreNet	79.68 / 66.51	68.04 / 50.77	61.66 / 47.55	56.14 / 37.83	59.97 / 43.84	66.92 / 46.59	65.40 / 48.84
	Ours	**80.11 / 66.84**	**69.04 / 51.20**	**64.58 / 49.75**	**58.54 / 41.03**	**62.77 / 46.54**	**67.92 / 47.19**	**67.16 / 50.44**

Table 4: The overall evaluation results (F1/EM) on the MLQA dataset.

Setting	Models	en	es	de	ar	hi	vi	Avg.
zero-shot	M-BERT	81.50 / 71.20	75.50 / 56.90	70.60 / 54.00	61.50 / 45.10	59.20 / 46.00	69.50 / 49.10	69.63 / 53.72
	XLM	81.30 / 68.80	75.60 / 56.90	72.60 / 55.50	62.60 / 43.20	63.10 / 46.00	70.40 / 48.70	70.93 / 53.18
	XLM-R$_{base}$	83.66 / 72.48	77.00 / 60.87	74.40 / 58.40	63.00 / 47.80	68.70 / 53.70	74.50 / 54.00	73.54 / 57.55
	Info-XLM	**85.15** / 72.80	76.15 / 59.30	73.88 / 59.00	63.51 / 49.78	69.66 / 54.90	73.21 / 55.25	73.76 / 58.51
	Ours	**84.51 / 74.59**	**78.25 / 61.67**	**75.89 / 59.79**	**65.18 / 50.04**	**70.79 / 55.45**	**75.74 / 56.46**	**75.06 / 59.87**
translate	XLM-R$_{base}$	82.59 / 71.30	78.55 / 60.20	76.42 / 60.69	65.15 / 48.42	71.35 / 56.43	76.10 / 56.68	75.03 / 58.96
	mixMRC	82.40 / 69.20	78.80 / 58.70	75.40 / 58.20	63.60 / 42.40	66.20 / 50.00	72.60 / 52.70	73.17 / 55,20
	LBMRC	83.40 / 70.10	80.00 / 59.60	76.50 / 59.80	65.00 / 44.50	67.40 / 52.00	74.60 / 55.50	74.48 / 56.92
	Ours	**84.06 / 73.11**	**80.04 / 61.68**	**77.88 / 62.48**	**66.54 / 50.34**	**73.77 / 58.90**	**77.64 / 57.49**	**76.66 / 60.66**

Table 5: The overall evaluation results (F1/EM) on the XQUAD dataset.

accuracy evaluation of MRC models. F1 measures the part of the overlapping mark between the predicted answer and the ground-truth answer. The exact match (EM) score is 1 if the prediction is exactly the same as the ground truth, otherwise 0.

Implementation Details

We build our model on XLM-R$_{base}$ on top of the Hugging Face Transformers[3], which contains 12 transformer layers. We use AdamW (Loshchilov and Hutter 2017) as our model optimizer, and the weight decay is set to 0.01 for both datasets. The learning rate is set to 3e-5. The size of $|\mathcal{Z}|$ and $|\mathcal{A}|$ are 20 and 50 in experiments, respectively[4]. During fine-tuning, we empirically set the max input sequence length to 384. The question max length is 64. We also use the warm-up proportion and set to 0.1. The τ in Eq.8 and batch size are 10 and 32, respectively. The α in Eq.9 is set to 0.5 in our experiments. We train the model using 8 NVIDIA V100 GPUs with 32 GB memory for each training language data with 8 epochs and save a checkpoint every 1000 steps.

Baselines

We compare our model with the following strong baselines: (1) M-BERT (Pires, Schlinger, and Garrette 2019), a cross-lingual version of BERT trained on 104 parallel languages and demonstrated highly competitive in multilingual language understanding tasks at zero- and few-shot settings; (2) XLM (Conneau and Lample 2019), another effective pre-trained multilingual language model achieving promising results on various cross-lingual tasks; (3) LAKM, a pre-trained task proposed by Yuan et al. (2020) via introducing extra parallel corpus for phrase level MLM; (4) mixMRC, a translation-based data augmentation strategy developed by Yuan et al. (2020) for xMRC; (5) LBMRC, a novel augmentation approach (Liu et al. 2020) based on knowledge distillation; (6) CalibreNet (Liang et al. 2021), a recent model aiming to enhance the boundary detection capability of PLMs in multilingual sequence labeling task; and (7) Info-XLM (Chi et al. 2021), a new state-of-the-art information-theoretic cross-lingual pre-training model. For fair comparisons, we use XLM-R$_{base}$ as our backbone architecture in this work.

Results

We compare our methods with the strong baselines in the two settings. The first is *zero-shot*: we fine-tune the state-of-the-art models in English only, and then test on the English and other five low-resource languages. The second is *translate-train*: we train the models by combining the translated data of all languages jointly during fine-tuning.

Results on MLQA In the first set of our experiments, we evaluate various baselines on the MLQA dataset, and the results are listed in Table 4. We make several observations from the results. First, our method outperforms all baselines

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

[4]Intuitively, with the size of $|\mathcal{A}|$ increasing, the model achieves better performance as it can mine more $hard$ negatives. Meanwhile, it also raises the computation cost. Empirically, we choose the $|\mathcal{A}|$ as 50 in our experiments.

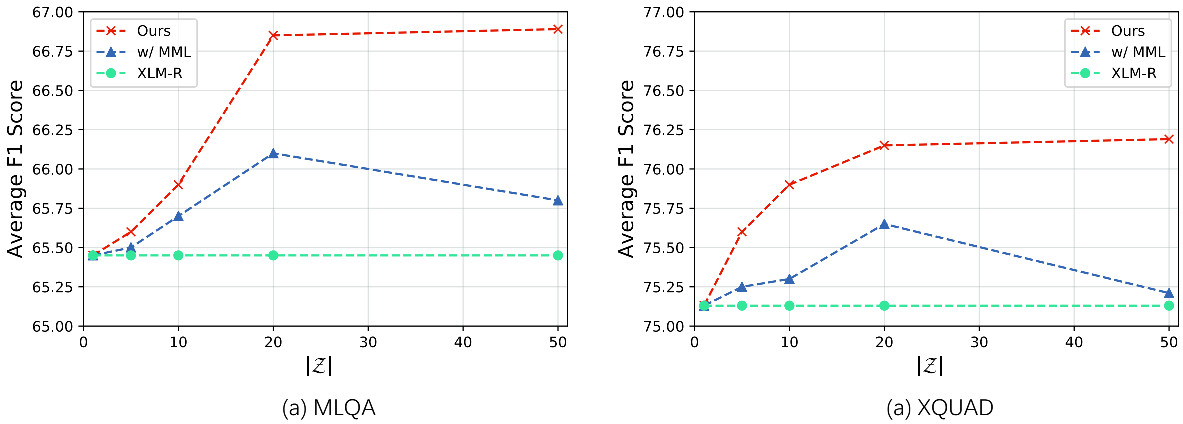

Figure 2: Model performances with different size of $\mathcal{Z}$ in training at the *translate-train* setting on both two datasets. We use the average F1 score of six languages as the evaluation metric.

in all languages at the *zero-shot* setting, indicating the effectiveness of our model. For instance, ours improves XLM-R$_{base}$ from 64.14% to 66.00% in F1 and from 46.00% to 48.88% in EM score on average. Moreover, in the *translate-train* setting, our approach achieves the best results 67.16% and 50.44% in F1/EM scores, respectively, which surpasses the strong baselines by a large margin. Third, compared with the LAKM and CalibreNet, which both utilize extra cross-lingual corpora, our model also obtains better results. Last, our model in the *zero-shot* setting even outperforms XLM-R$_{base}$ in the *translate-train* setting. This confirms the effectiveness of the proposed Hard-Learning algorithm and Answer-Aware Contrastive Learning.

Results on XQUAD In order to show the generality, we also evaluate our approach on other common used xMRC benchmark called XQUAD in six languages. The experimental results are reported in Table 5, which are also under the *zero-shot* and *translate-train* settings. Clearly, our method consistently outperforms the strong baselines in both settings. Specifically, our best model outperforms XLM-R$_{base}$ in the *translate-train* setting with a clear margin in both F1 and EM scores. In the *zero-shot* setting, our model also obtains on average 1.52% and 2.32% improvement F1 and EM scores, respectively, in those languages. Even compared with other strong baselines like mixMRC and LBMRC, ours also show its superiority. The evaluation results on XQUAD further verify the effectiveness and robustness of our method.

Analysis

In this section, we conduct a series of ablation studies and analysis to better understand what contributes to the performance advantages of our model. Furthermore, we present the ablation study of hyper-parameter τ and AA-CL in Appendix A and B.

Models	es	ar	vi
Ours	69.04/51.20	58.54/41.03	67.92/47.19
- *HL*	67.64/49.45	57.10/39.18	66.00/45.49
- *AA-CL*	67.70/50.12	57.66/39.80	67.00/46.07
w/ *MML*	68.47/50.21	57.46/40.00	66.89/46.01

Table 6: Ablation study of our methods on MLQA dataset at *translate-training* setting. We evaluate each method in three languages: *Spanish, Arabic and Vietnamese.*

Key Components

To evaluate the effectiveness of our model, we conduct ablation studies by removing each key component individually. As shown in Table 6, there is a obvious performance gap when removing *HL*, indicating that pre-obtaining a set of predictions and training a model through hard updates play an important part in performance. Then, removing *AA-CL*, the model performance drops inevitably. The results demonstrate the effectiveness of this coarse-to-fine method for utilizing hard-negatives from high confident predictions over training. In general, each key component contributes to the performance improvement of the model. In Table 6 we provide the results using MML as our training objective. The model performance drops about 1% in F1 and EM scores on three languages, indicating the effectiveness of HL algorithm once again.

Size of $\mathcal{Z}$

To assess how the proposed hard learning algorithm works with respect to the size of pre-obtained predictions set ($|\mathcal{Z}|$), we conduct a series of experiments on both datasets with $|\mathcal{Z}| = \{1, 5, 10, 20, 50\}$. Figure 2 shows the results. For fair comparisons, AA-CL is removed in this experiment. Figure 2 shows that our proposed method outperforms MML and the baseline consistently with different values of $|\mathcal{Z}|$. When $|\mathcal{Z}|$ is set to 20 and 50, the model achieves comparable performances on the two datasets. Considering the computation

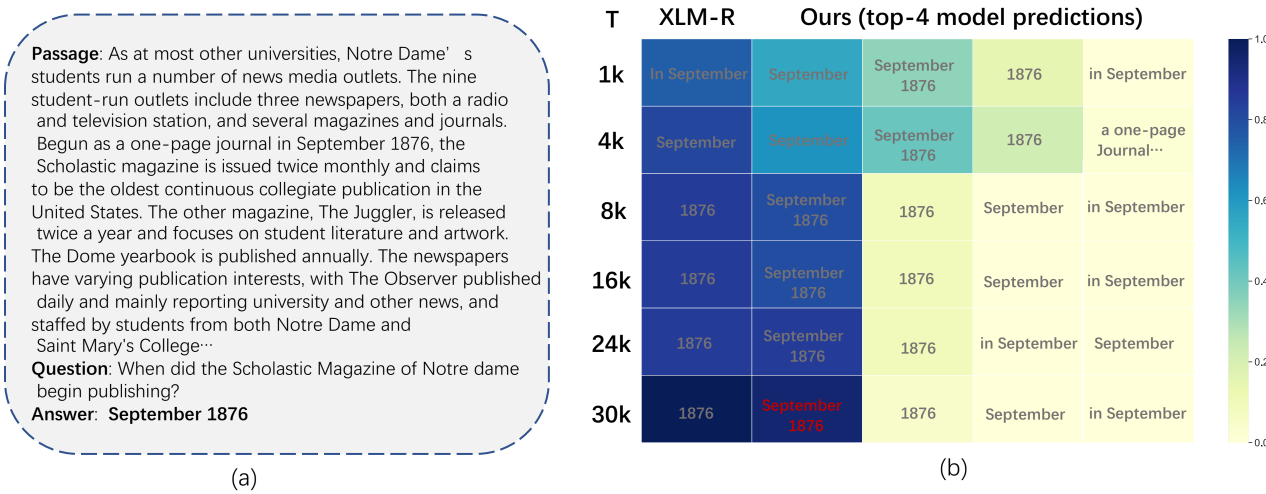

Figure 3: An example from MLQA dataset, with its ground-truth answer "September 1876". For each iteration step T, we present top-1 prediction from the baseline (XLM-R_{base}) and top 4 predictions from ours at the *translate-train* setting.

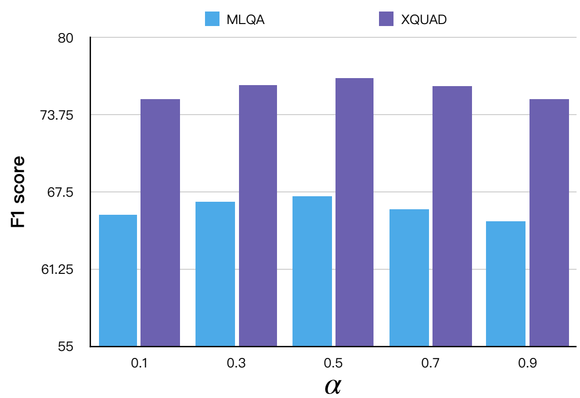

Figure 4: Model performances with different α in training at the *translate-train* setting. We use the average F1 score of six languages as the evaluation metric.

efficiency and memory cost, we choose $|\mathcal{Z}| = 20$ in our main experiments.

Case Study of Model Predictions Over Training

To show how our model performs during the training process, we analyze the top predictions and assigned likelihood from the models with respect to different iteration steps (from 1k to 30k). Figure 3 shows that both the baseline and our model first begin by assigning higher probabilities to wrong predictions, like "in September" and "September", but gradually our method leans to favor the true prediction. Unfortunately, XLM-R_{base} still insists on making the wrong prediction until the end of the training, indicating that it may confused by the similar spans with the correct answer ("1876" vs. "September 1876"), which can be seen as an understandable mistake. The visualization in Figure 3 (b) shows the ability of our model in identifying the correct answer from many similar spans.

Hyper-parameter α

It is essential to study the sensitivity analysis of α, since we train our model in a multi-task manner. Thereafter, we conduct additional experiments to study the effect of different values of α on optimizing the model on both datasets. We test the model performance with $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. From Figure 4, we find that the model performances on MLQA and XQUAD show similar trends α, and our method achieves the best results while $\alpha = 0.5$.

Conclusion

In this paper, we tackle the challenge of exploring the potential of mining useful task-related knowledge from n-best answer predictions. Concretely, we decompose the training for xMRC model into two stages: (1) At the first stage, we target at recall at top-k predicted results, and thus, develop a hard learning algorithm to progressively encourage the model to give higher attention to the pre-obtained top-k predictions with taking these as weak supervision. (2) Then, we propose an answer-aware contrastive learning to strengthen the model's ability to further distinguish the correct span from top-k possible spans to achieve the goal of precision at top-1. The experimental results show that our model achieves competitive performances compared to the state-of-the-art on two public benchmark datasets. The systematic analysis further demonstrates the effectiveness of each component in our model. Future work can include an extension of how to employ AA-CL to other natural language understanding tasks.

Acknowledgments

Jian Pei's research is supported in part by the NSERC Discovery Grant program. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Asai, A.; Eriguchi, A.; Hashimoto, K.; and Tsuruoka, Y. 2018. Multilingual Extractive Reading Comprehension by Runtime Machine Translation. *CoRR*, abs/1809.03275.

Chen, N.; Liu, F.; You, C.; Zhou, P.; and Zou, Y. 2021. Adaptive Bi-Directional Attention: Exploring Multi-Granularity Representations for Machine Reading Comprehension. In *ICASSP*, 7833–7837. IEEE.

Chi, Z.; Dong, L.; Wei, F.; Yang, N.; Singhal, S.; Wang, W.; Song, X.; Mao, X.; Huang, H.; and Zhou, M. 2021. InfoXLM: An Information-Theoretic Framework for Cross-Lingual Language Model Pre-Training. In *NAACL-HLT*, 3576–3588. Association for Computational Linguistics.

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Unsupervised Cross-lingual Representation Learning at Scale. *CoRR*, abs/1911.02116.

Conneau, A.; and Lample, G. 2019. Cross-lingual Language Model Pretraining. In *NeurIPS*, 7057–7067.

Cui, Y.; Che, W.; Liu, T.; Qin, B.; Wang, S.; and Hu, G. 2019. Cross-Lingual Machine Reading Comprehension. In *EMNLP/IJCNLP (1)*, 1586–1595. Association for Computational Linguistics.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *CoRR*, abs/2104.08821.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *CVPR (2)*, 1735–1742. IEEE Computer Society.

Huang, H.; Liang, Y.; Duan, N.; Gong, M.; Shou, L.; Jiang, D.; and Zhou, M. 2019. Unicoder: A Universal Language Encoder by Pre-training with Multiple Cross-lingual Tasks. In *EMNLP/IJCNLP (1)*, 2485–2494. Association for Computational Linguistics.

Kruengkrai, C.; Nguyen, T. H.; Mahani, S. A.; and Bing, L. 2020. Improving Low-Resource Named Entity Recognition using Joint Sentence and Token Labeling. In *ACL*, 5898–5905. Association for Computational Linguistics.

Lewis, P. S. H.; Oguz, B.; Rinott, R.; Riedel, S.; and Schwenk, H. 2020. MLQA: Evaluating Cross-lingual Extractive Question Answering. In *ACL*, 7315–7330. Association for Computational Linguistics.

Liang, S.; Shou, L.; Pei, J.; Gong, M.; Zuo, W.; and Jiang, D. 2021. CalibreNet: Calibration Networks for Multilingual Sequence Labeling. In *WSDM*, 842–850. ACM.

Liang, Y.; Duan, N.; Gong, Y.; Wu, N.; Guo, F.; Qi, W.; Gong, M.; Shou, L.; Jiang, D.; Cao, G.; Fan, X.; Zhang, R.; Agrawal, R.; Cui, E.; Wei, S.; Bharti, T.; Qiao, Y.; Chen, J.; Wu, W.; Liu, S.; Yang, F.; Campos, D.; Majumder, R.; and Zhou, M. 2020. XGLUE: A New Benchmark Datasetfor Cross-lingual Pre-training, Understanding and Generation. In *EMNLP (1)*, 6008–6018. Association for Computational Linguistics.

Liu, J.; Shou, L.; Pei, J.; Gong, M.; Yang, M.; and Jiang, D. 2020. Cross-lingual Machine Reading Comprehension with Language Branch Knowledge Distillation. In *COLING*, 2710–2721. International Committee on Computational Linguistics.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Min, S.; Chen, D.; Hajishirzi, H.; and Zettlemoyer, L. 2019. A Discrete Hard EM Approach for Weakly Supervised Question Answering. In *EMNLP/IJCNLP (1)*, 2851–2864. Association for Computational Linguistics.

Pires, T.; Schlinger, E.; and Garrette, D. 2019. How multilingual is Multilingual BERT? *arXiv preprint arXiv:1906.01502*.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *ACL (2)*, 784–789. Association for Computational Linguistics.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *EMNLP*, 2383–2392. The Association for Computational Linguistics.

Reddy, S.; Chen, D.; and Manning, C. D. 2019. CoQA: A Conversational Question Answering Challenge. *Trans. Assoc. Comput. Linguistics*, 7: 249–266.

Seo, M. J.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2017. Bidirectional Attention Flow for Machine Comprehension. In *ICLR (Poster)*. OpenReview.net.

Singh, J.; McCann, B.; Keskar, N. S.; Xiong, C.; and Socher, R. 2019. XLDA: Cross-Lingual Data Augmentation for Natural Language Inference and Question Answering. *CoRR*, abs/1905.11471.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1): 1929–1958.

Wu, Z.; Wang, S.; Gu, J.; Khabsa, M.; Sun, F.; and Ma, H. 2020. CLEAR: Contrastive Learning for Sentence Representation. *CoRR*, abs/2012.15466.

You, C.; Chen, N.; Liu, F.; Yang, D.; and Zou, Y. 2020. Towards Data Distillation for End-to-end Spoken Conversational Question Answering. *CoRR*, abs/2010.08923.

Yu, A. W.; Dohan, D.; Luong, M.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In *ICLR (Poster)*. OpenReview.net.

Yuan, F.; Shou, L.; Bai, X.; Gong, M.; Liang, Y.; Duan, N.; Fu, Y.; and Jiang, D. 2020. Enhancing Answer Boundary Detection for Multilingual Machine Reading Comprehension. In *ACL*, 925–934. Association for Computational Linguistics.

Zhou, Y.; Geng, X.; Shen, T.; Zhang, W.; and Jiang, D. 2021. Improving Zero-Shot Cross-lingual Transfer for Multilingual Question Answering over Knowledge Graph. In *NAACL-HLT*, 5822–5834. Association for Computational Linguistics.