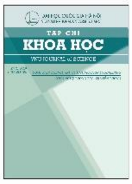
VNU Journal of Science:
Computer Science and Communication Engineering
Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce



Original Article

ViMRC - VLSP 2021: An empirical study of Vietnamese Machine Reading Comprehension with Unsupervised Context Selector and Adversarial Learning

Tran Hoang Vu*, Nguyen Phúc Minh*

Vinbrain, 458 Minh Khai, Hai Ba Trung, Hanoi, Vietnam

Received 19 December 2021
Revised 18 April 2022; Accepted 5 May 2022

Abstract: Machine Reading Comprehension (MRC) is a great NLP task that requires concentration on making the machine read, scan documents, and extract meaning from the text, just like a human reader. one of the MRC system challenges is not only having to understand the context to extract the answer but also being aware of the trust-worthy of the given question is possible or not. Thought pre-trained language models (PTMs) have shown their performance on many NLP downstream tasks, but it still has a limitation in the fixed-length input. We propose an unsupervised context selector that shortens the given context but still contains the answers within related contexts. In VLSP2021-ViMRC Challenge [1] dataset, we also empirical several training strategies consisting of unanswerable question sample selection and different adversarial training approaches, which slightly boost the performance 2.5% in EM score and 1% in F1 score.

Keywords: Machine reading comprehension, Adversarial learning, Vietnamese.

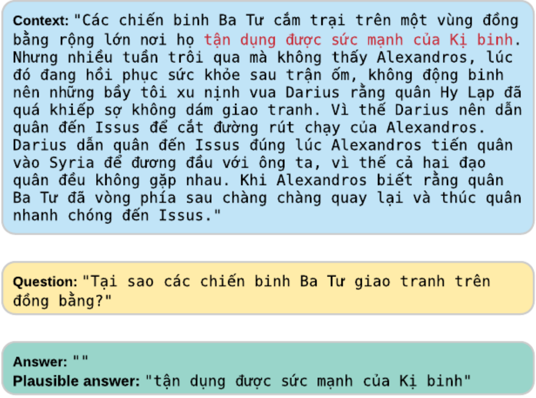
1. Introduction

Machine Reading Comprehension (MRC) is a task introduced to test the level at which a machine can understand natural languages by asking the machine to answer questions based on a given context. The early MRC systems were designed on a latent hypothesis that all questions can be answered according to a given context, which is not always true for real-world cases. The current MRC task has required that the model have to classify unanswerable and answerable questions to avoid giving plausible answers. Figure 1 shows an unanswerable example from UIT-ViQuAD 2.0 dataset [1]. PTMs such as ELMo [2], GPT [3], or BERT [4] have been proposed and achieved superior results on MRC tasks by capturing contextual representation features. Most of BERT-family architecture [4] usually face to the limitation of fixed input-length. This make a long input must be partitioned into smaller segments of manageable sizes and leads to the loss of salient cross-segment information, the context

[*] Corresponding author.
E-mail address: v.vutran@vinbrain.net; v.vutran@vinbrain.net

https://doi.org/10.25073/2588-1086/vnucsce.356

fragmentation problem. Although researchers [5, 6] proposed new architecture to solve this problem, these previous works are focus on English only. Inspired by selecting salient sentences before extract the span answer [7, 8], we introduce an unsupervised context selector that address the long input context.



Context: "Các chiến binh Ba Tư cắm trại trên một vùng đồng bằng rộng lớn nơi họ tận dụng được sức mạnh của Kị binh. Nhưng nhiều tuần trôi qua mà không thấy Alexandros, lúc đó đang hồi phục sức khỏe sau trận ốm, không động binh nên những bầy tôi xu nịnh vua Darius rằng quân Hy Lạp đã quá khiếp sợ không dám giao tranh. Vì thế Darius nên dẫn quân đến Issus để cắt đường rút chạy của Alexandros. Darius dẫn quân đến Issus đúng lúc Alexandros tiến quân vào Syria để đương đầu với ông ta, vì thế cả hai đạo quân đều không gặp nhau. Khi Alexandros biết rằng quân Ba Tư đã vòng phía sau chàng chàng quay lại và thúc quân nhanh chóng đến Issus."

Question: "Tại sao các chiến binh Ba Tư giao tranh trên đồng bằng?"

Answer: " "
Plausible answer: "tận dụng được sức mạnh của Kị binh"

Figure 1. An unanswerable MRC example in the VLSP2021-ViMRC challenge dataset. The highlighted span text in context is the plausible answer for the question.

Adversarial training (AT) [9] is a means of regularizing classification algorithms by generating adversarial noise to the training data. In the Machine Reading Comprehension task, AT has been leveraged for learning domain-invariant representation [10], which made the MRC model generalize well to predict answers on unseen out-of-domain. The performance of models also has been shown the improvement while applying Virtual Adversarial Training (author?) on SQuAD1.1 [12], SQuAD2.0 [13] and RACE[14]. According to the benefits of AT, we decided to apply several training strategies that can boost the model performance across MRC tasks which is discussed further in Section 2.3 and Section 2.4.

Our contributions are summarized as follows:

- We introduce an unsupervised context selector to solve the long context problem.
- We introduce a simple strategy to generate unanswerable examples, called Question-Context Shuffle.
- We experiment with different adversarial training approaches in MRC.

We evaluate and experiment with the proposed methods on the dataset released by VLSP2021-ViMRC Challenge [1].

2. Background

2.1. MRC for Vietnamese

There are limited studies of understanding a text and answering relevant questions for Vietnamese. Most of the Vietnamese dataset that is close to the MRC task comes from AI challenge or Shared-task workshop[1]. It makes lack of benchmark datasets for Vietnamese to develop robust PTMs or MRC models. Then for the first time, UIT-ViQuAD[15] is the first public academic dataset in MRC for Vietnamese. The author benchmarked the dataset in different embeddings, and multilingual PTMs showed the best performance on this dataset.

2.2. Pre-trained Language Models

PTMs on the large unlabeled corpus has shown impressive performance on many downstream NLP tasks, proving that they can learn universal patterns. There have been several applications for using pre-trained language models that can capture contextual word embeddings, such as ELMo [2], GPT [3], or BERT [4] to transfer the knowledge from pre-training to various downstream tasks.

For the very first time that BERT has been introduced, it significantly outperforms previous SOTA models on eleven NLP tasks in GLUE [16]. In terms of monolingual language models pre-trained for Vietnamese, it has shown significant improvements in Named Entity Recognition, Parsing, and Natural Language Inference tasks. PhoBERT pre-training approach is based on RoBERTa [18] which optimizes the BERT pre-training procedure for more robust performance.

Given input con text sequence $C = \{c_1, c_2, ..., c_N\}$ and question $Q = \{q_1, q_2, ..., q_M\}$ where N is the context length and M is the question length. The model has to verify the question is answerable or not, for each answerable predictions, the model is enabled to output the correct answer span. The answer span A is either a valid span $A = \{a_i, a_2, ..., a_j\}$ where $1 \leq i \leq j \leq N$ or an empty $A = \{\}$. The input model is the concatenation of C and Q with special tokens [CLS] and [S E P] as [CLS] Q [S E P] C [S E P]. We employ a linear layer with Softmax operation and feed last-layer hidden representation H ∈ RLXd as the input to obtain the start/end position probability distributions ps, pe respectively. The training objective of answer span prediction is defined as cross entropy loss for the start and end index position.

$$loss_{start/end_{idx}} = -\frac{1}{N_k}\sum_{k}^{N_k}[y_s^k log(p_s^k)+y_e^k log(p_e^k)] \quad (1)$$

where N_k is the number of examples, y_s^k and y_e^k are respectively ground-truth start and end position of example k. We also employ linear layer with Softmax for h_{CLS} ∈ H and use cross entropy as loss function for classification answerable/unanswerable question.

$$loss_{CLS} = -\frac{1}{N_k}\sum_{k}^{N_k}\sum_{u}^{U}[y_u^k log(p_u^k)] \quad (2)$$

where p_u^k is answerable and unanswerable probability distributions. U means the number of classes (U = 2 in this work). The overview of our method architecture is illustrated in Figure 2.

2.3. Adversarial Training

Small perturbations to the input images can mislead models to predict wrong labels in the image classification, and the perturbed inputs are called adversarial examples [19]. Then, a simple adversarial training method has been proved can improve the robustness of the model by training on both clean examples and adversarial examples [9]. In NLP tasks, a popular approach to generate perturbations is to perturb word vectors from the embedding layer. In general, adversarial training idea is formulated as follows:

$$y = f_\theta(x) \quad (3)$$
$$y' = f_\theta(x + noise) \quad (4)$$

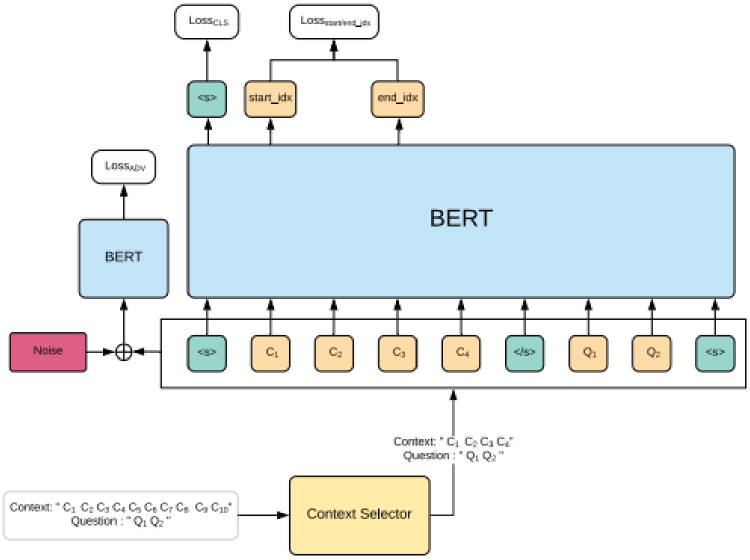


Figure 2. The overview architecture of our method.

where θ is our model weight, x is the embedding of the input sequence and noise is simply a tensor that is randomly generated with normal distribution. Motivated by making the MRC model more generalized with diverse inputs, we apply adversarial learning, which is a noise layer for the input. In this work, we utilize R3F [20] that encourages the model to generalize with representation changes during training without hurting performance. The adversarial training loss $loss_{ADV}$ is calculated by the following:

$$loss_{ADV} = KL(y, y') + KL(y, y')$$

where KL is the KL-Divergence. The final loss function is the summation of mentioned loss with λ0, λ1 and λ2 are learned weight for each task:

$$loss = \lambda_0 loss_{CLS} + \lambda_1 loss_{start/end_{idx}} + \lambda_2 loss_{ADV} \tag{6}$$

2.4. Domain Agnostic (DA)

Adapting models to a new domain without fine-tuning is a challenging problem in deep learning. In this paper, we also experiment with adversarial training called Domain-agnostic. The adversarial training is leveraged for learning domain-invariant representation. Specifically, the MRC model learns to make the discriminator that classifies the joint embedding of context and question into the given T domains. If the discriminator cannot tell the difference between embeddings from different T domains, the MRC model learns domain-invariant feature representation.

The discriminator is trained to minimize the KL divergence between uniform distribution over T classes and discriminator's prediction:

$$loss_{ADV} = -\frac{1}{N}\sum_{t=1}^{T}\sum_{k=1}^{N_k} KL(U(l) \| P(l_t^k | h_t^k)) \tag{7}$$

where l is domain category, U(l) is the uniform distribution over T classes and h is the hidden representation of both context and question. N_k is number of samples of class k and N is total samples.

3. Method

3.1. Unsupervised Context Selector

Due to the input sequence may exceeding the beneficial length of BERT [4] (256 tokens), the losing context results in not only a missing answer context but also harm the model by learning a noisy sample. We introduce an unsupervised context selector that shortens the context but still contains the answer within related contexts. The context selector takes context and question as input then outputs a shorter version of the context while ensuring the answer must be included. We observe that almost all of the questions focus on the entities in the question, so we want to take advantage of these properties to shorten the context.

Since the linguistic style and syntactic of both context and question from the dataset are formal, we decided to use POS-TAGER from underthesea which has been trained on a dataset that has a similar distribution of the former dataset. Given the question, we filter stopwords and use POS-TAGER from underthesea to get POS output. Then we select important phrases based on the following output with tags: 'N','Np','V','Vp' to finalize a phrase set N. The context is chunked by sentence segmentation from NLTK [21], each sentence is scored by the occurrence of tokens that are included in the extracted phrases. The sentence s has t syllable-level tokens would be selected if it has a score score(s) > 2ϵ as following:

$$score(s) = max(f(s) + f(s+1); f(s) + f(s-1)) \tag{8}$$

$$f(s) = \sum_{t \epsilon N} g(t) \tag{9}$$

where $\epsilon = \sum_{t \in s; score(t) \# 0} score(t)$, $g(t)$ is the number of co-occurrence of an token t in the given context and question. We also select the previous and next sentence of the selected sentence to make a leading sentence and augment the surrounding context.

3.2. Question-Context Shuffle

According to Table 3, there is an imbalance between answerable and unanswerable questions. This makes the model easily predict plausible answers and mistaken the given context and question. We introduce a simple strategy called Question-Context shuffle to generate unanswerable examples from the training set. This approach aims to augment more unanswerable samples by getting a random irrelevance question for each given context.

We divide the generated unanswerable samples into two types are $EXAMPLES_{hard}$ and $EXAMPLES_{easy}$. for each context, the selected unanswerable questions are from different context but the sample title are categorized into

EXAMPLES$_{hard}$ while the unanswerable questions from different title are categorized into EXAMPLES$_{easy}$. The title is the main topic of many contexts which is shown in Table 1. The statistic of the dataset after pre-processing is presented in Table 2 in which the total samples of two class has been balanced.

Table 1. Sample structure of UIT-ViQuAD 2.0 dataset. Each example contains a title corresponds to the the category of the context. Each example has lots of context, each context contains multiple (question,answer,is-possible) triplets

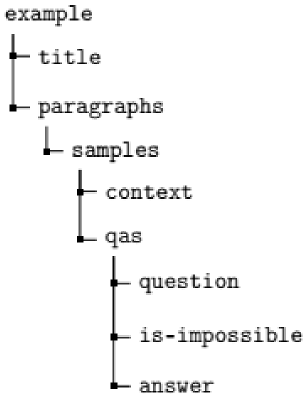
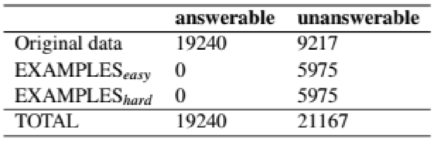


Table 2. Statistic of classes of training datasetafter data augmentation



	answerable	unanswerable
Original data	19240	9217
EXAMPLES$_{easy}$	0	5975
EXAMPLES$_{hard}$	0	5975
TOTAL	19240	21167

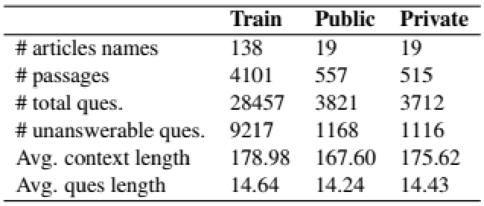
4. Experimental Results

4.1. Setup

We employ RDRSegmenter [22] from VnCoreNLP [23] to perform word-level and sentence segmentation on UIT-ViQuAD 2.0 dataset (e.g "Những cá_thể xung_quanh ghi_nhớ tôm tít bằng cách nào ?"). Our experimental models were implemented PyTorch [24] and utilize Huggingface's Transformers [25] for pretrained language models. In our experiments, almost all experiments used the Shuffle-Context Shuffle strategy to make to model aware of more data.

In practice, we have three-phase of training. In the first phase, we make the model generalize with and warm up with the data by setting the λ0 = 0.2, λ1 = 0.6, and λ2 = 0.2. We observed that the loss$_{ADV}$ is converged after the first phase, we decided to set λ2 = 0 on every next phase. In the second phase, we aim to make the classification loss which only saves the checkpoint with the lowest loss on the dev set. In the third phase, we focus on the start/end index loss, which considers only the best checkpoint based on CE loss of start/end on dev set. We set the λ0 = 0.9 and λ1 = 0.1 on the second phase and λ0 = 0.1 and λ1 = 0.9 on the third phase.

Table 3. Data analysis of UIT-ViQuAD 2.0 dataset. # stands for numbers of samples. **Public**stands for Public testset. **Private** stands for Private testset. The average length unit is calculated in syllable-level



	Train	Public	Private
# articles names	138	19	19
# passages	4101	557	515
# total ques.	28457	3821	3712
# unanswerable ques.	9217	1168	1116
Avg. context length	178.98	167.60	175.62
Avg. ques length	14.64	14.24	14.43

4.2. Dataset

In VLSP2021-ViMRC Challenge [1], the dataset is organized into 3 sets are train/public test/private test has 138/19/19 number of articles respectively. The analysis of the dataset is shown in Table 3. Since there is no dev set, we decided to categorize the articles in the training dataset into two main sets based on answerable and unanswerable questions, making the split dataset balanced in categories and no leaked articles. Then we randomly split these two sets with a ratio of 9/1 before uniting them into a train/dev set based on the mentioned ratio.

4.3. Hyperparameters

In all experiment settings, we use Adam optimizer [26] with a learning rate of 1e-5 without warm-up steps, batch size of 32. In the inference stage, we set the threshold δ is 0.4 to determine if the question is answerable or not. We set the maximum sequence length for context and questions to be 230 and 50 for each sample. All experiments are launched with a maximum of 10 epochs and a single A100-40GB GPU device.

4.4. Metric

We evaluated our models by Exact Match (EM) and F1 score for each question-answer pair. The EM score measures the percentage of predictions that match ground-truth answers in character level. The F1 metric aims to care equally about precision and recall of the number of shared words between the prediction and the truth. The precision is the ratio of the number of shared words to the total number of words in the prediction, while recall is the ratio of the number of shared words to the total number of words in the ground truth. The higher the F1 and EM scores are, the closer ground truth and predicted context is.

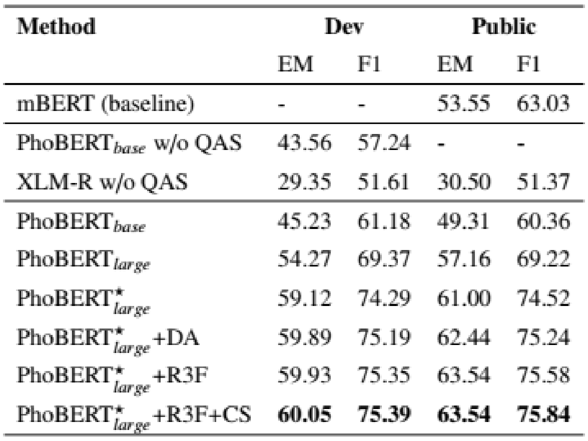
4.5. Results

4.5.1. Main Result

We use two main PTMs as backbone are: PhoBERT that supports a maximum of 256 tokens, and XLM-Roberta that provides a maximum input length is 512 tokens. We observed that monolingual models (e.g phoBERT [17]) perform better than multilingual models (e.g mBERT [4], XLM-Roberta [27]). Moreover, training monolinguals on a word-level dataset improves performance significantly due to improved quality of words and reduced length of context and question pairs. Using methods Context Selector and Adversarial Training also slightly improve performance. Result experiment is shown on Table 4.

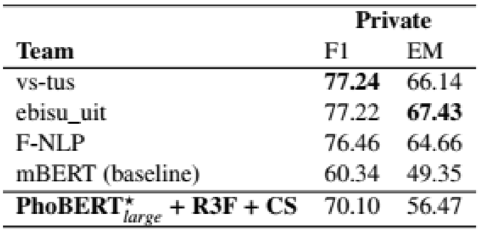
In terms of the private test set, our method has exceeded the baseline +9.76 in F1 score and +7.12 in Exact Match score. However, our method still shows limitations compare to top-3 teams and we would discuss them in Section 4.6. The result of the top-3 teams and our result in the private test is illustrated in Table 5.

Table 4. Results on the UIT-ViQuAD 2.0 public test set. (R3F, DA) refers to adversarial training methods. (CS) refers to Context Selector. ⋆ refers to word-level. w/o QAS refers to without Question-Context shuffle



Method	Dev		Public	
	EM	F1	EM	F1
mBERT (baseline)	-	-	53.55	63.03
PhoBERT$_{base}$ w/o QAS	43.56	57.24	-	-
XLM-R w/o QAS	29.35	51.61	30.50	51.37
PhoBERT$_{base}$	45.23	61.18	49.31	60.36
PhoBERT$_{large}$	54.27	69.37	57.16	69.22
PhoBERT$^{\star}_{large}$	59.12	74.29	61.00	74.52
PhoBERT$^{\star}_{large}$+DA	59.89	75.19	62.44	75.24
PhoBERT$^{\star}_{large}$+R3F	59.93	75.35	63.54	75.58
PhoBERT$^{\star}_{large}$+R3F+CS	**60.05**	**75.39**	**63.54**	**75.84**

Table 5. Results on UIT-ViQuAD 2.0 private testset. ⋆ refer to word-level



	Private	
Team	F1	EM
vs-tus	**77.24**	66.14
ebisu_uit	77.22	**67.43**
F-NLP	76.46	64.66
mBERT (baseline)	60.34	49.35
PhoBERT$^{\star}_{large}$ + R3F + CS	70.10	56.47

4.5.2. Context Selector

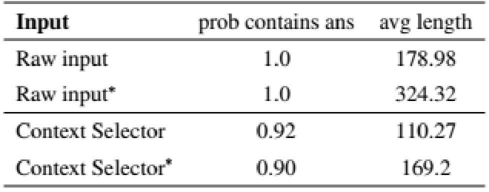
We also evaluate our unsupervised Context Selector on the train set, which is shown in Table 6. The probability that the shortened context contains an answer shows competitive results compared to the raw input. In terms of the average context length, the Context Selector helps the model to receive salient sentences only by reducing from 324,32 tokens to 169,2 tokens. The result shows that the Context Selector has crucially reduced the context length while retaining the answer in the filtered context.

4.6. Error Analysis

We also examined the errors of our method in the dev dataset that decrease the evaluation score significantly. The major errors are:

Span error: We found that about 40% of errors are span errors. More specifically, the start and end index from the model prediction usually is shifted from the correct ground truth. We hypothesis that this span error may come from the annotator's bias. It is difficult for the model to be aware of samples with ambiguous answer text. Table 7 shows a few span error examples that we have analyzed in VLSP2021-ViMRC Challenge.

Table 6. Results of Context Selector on T-ViQuAD 2.0 train set. * refers samples that has context length > 256 syllable tokens

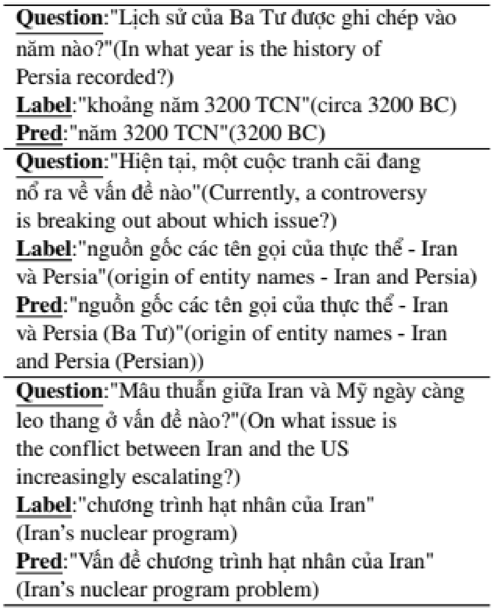


Input	prob contains ans	avg length
Raw input	1.0	178.98
Raw input*	1.0	324.32
Context Selector	0.92	110.27
Context Selector*	0.90	169.2

Misclassify answerable/unanswerable: About 35% of errors are failures of misclassifying answerable and unanswerable questions. According to our experiment on dev set, the best threshold δ to classify either the answerable question or not is 0.4. It means that our model does not generalize for the classification of the question when encountering out-of-domain questions.

Context Selector: Since we use the context selector to shorten the context length for each input sequence, the performance of the whole pipeline still depends on the context selector output result. We observe that the context selector dealt with straightforward questions well (e.g., "Tên của vua Nam_Hán là gì?"). However, it has two main drawbacks not exploiting the training data and depending on manual rules. This makes the context selector unable to acknowledge the entities in the dataset domain and has a limited ability to handle multi-hop questions. Moreover, the surrounding context of the answer may not be sufficient or related to the filtered context, which may hurt the model on prediction.

Table 7. Examples of error analysis in VLSP-2021 MRC. **Label** refers to Grouth-truth of the question. **Pred** refers to predictions of the model with given question



Question:"Lịch sử của Ba Tư được ghi chép vào năm nào?"(In what year is the history of Persia recorded?)
Label:"khoảng năm 3200 TCN"(circa 3200 BC)
Pred:"năm 3200 TCN"(3200 BC)

Question:"Hiện tại, một cuộc tranh cãi đang nổ ra về vấn đề nào"(Currently, a controversy is breaking out about which issue?)
Label:"nguồn gốc các tên gọi của thực thể - Iran và Persia"(origin of entity names - Iran and Persia)
Pred:"nguồn gốc các tên gọi của thực thể - Iran và Persia (Ba Tư)"(origin of entity names - Iran and Persia (Persian))

Question:"Mâu thuẫn giữa Iran và Mỹ ngày càng leo thang ở vấn đề nào?"(On what issue is the conflict between Iran and the US increasingly escalating?)
Label:"chương trình hạt nhân của Iran" (Iran's nuclear program)
Pred:"Vấn đề chương trình hạt nhân của Iran" (Iran's nuclear program problem)

5. Conclusion

We introduce applied Context Selector to overcome the large context problem, which is a major limitation of PTMs. We introduce Question-Passage shuffle to solve imbalanced data by generating unanswerable examples. In addition, we investigated the effect of some adversarial training methods on the VLSP2021-ViMRC Challenge dataset. We also show error analysis which helps future studies in MRC or interested research utilize our method. Our experiments demonstrate that adversarial training methods improve the MRC model over the pre-trained model 1%.

References

[1] N. V. Kiet, T. Q. Son, N. T. Luan, H. V. Tin, L. T. Son, N. L. T. Ngan, VLSP 2021 - ViMRC Challenge: Vietnamese Machine Reading Comprehension, *VNU Journal of Science:*

Computer Science and Communication Engineering, Vol. 38, No. 2, 2022.

[2] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep Contextualized Word Representations, In: Proc. of Naacl, 2018.

[3] V. Cohen, A. Gokaslan, Opengpt-2: Open Language Models and Implications of Generated Text, Xrds, Vol. 27, No. 1, 2020, pp. 26–30. Doi:10.1145/3416063.

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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, Vol. 1, 2019, pp. 4171–4186. Doi:10.18653/V1/N19-1423.

[5] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, R. Salakhutdinov, Transformer-Xl: Attentive Language Models Beyond a Fixed-LengthContext, In: Proceedings of The 57th Annual Meeting of The Association for Computational Linguistics, Florence, Italy, 2019, pp. 2978–2988. Doi:10.18653/V1/P19-1285.

[6] S. Ding, J. Shang, S. Wang, Y. Sun, H. Tian, H. Wu, H. Wang, Ernie-Doc: A Retrospective Long-Document Modeling Transformer, In: Proceedings of the 59th Annual Meeting of The Association for Computational Linguistics and The 11th International Joint Conference on Natural Language Processing, Vol. 1, 2021, pp. 2914–2927.

[7] S. Gehrmann, Y. Deng, A. Rush, Bottom-Up Abstractive Summarization, In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4098–4109.

[8] M. P. Nguyen, N. T. Tran, Improving Abstractive Summarization with Segment-Augmented and Position-Awareness, Procedia Computer Science, Vol. 189, 2021, pp.167–174

[9] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and Harnessing Adversarial Examples, Corr Abs/1412.6572.

[10] S. Lee, D. Kim, J. Park, Domain-Agnostic Question-Answering with Adversarial Training, In:Emnlp, 2019.

[11] Z. Yang, Y. Cui, W. Che, T. Liu, S. Wang, G. Hu, Improveing Machine Reading Comprehension Via Adversarial Training, Arxiv Abs/1911.03614.

[12] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, Squad: 100,000+ Questions for Machine Comprehension of Text, In: Proceedings of The 2016 Conference on Empirical Methods In Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 2383–2392.

[13] P. Rajpurkar, R. Jia, P. Liang, Know What You Don't Know: Unanswerable Questions for Squad, In: Proceedings of The 56th Annual Meeting of The Association for Computational Linguistics, Vol. 2,2018, pp. 784–789. Doi:10.18653/V1/P18-2124.

[14] Z. Yang, Y. Cui, C. Si, W. Che, T. Liu, S. Wang, G. Hu, Adversarial Training for Machine Reading Comprehension with Virtual Embeddings, In: Proceedings of *Sem 2021: The Tenth Joint Conference on Lexical And Computational Semantics, Association for Computational Linguistics, 2021, pp. 308–313.

[15] K. Nguyen, V. Nguyen, A. Nguyen, N. Nguyen, A Vietnamese Dataset for Evaluating Machine Reading Comprehension, In: Proceedings of The 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, 2020, pp. 2595–2605.

[16] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, Glue: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding, in: Proceedings of The 2018 Emnlp Workshop Blackboxnlp: Analyzing And Interpreting Neural Networks for Nlp, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 353–355. Doi:10.18653/V1/W18-5446.

[17] D. Q. Nguyen, A. T. Nguyen, Phobert: Pre-Trained Language Models for Vietnamese, In: Findings of The Association for Computational Linguistics: Emnlp 2020, 2020, pp. 1037–1042.

[18] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A Robustly Optimized Bert Pretraining Approach, 2019, Cite Arxiv:1907.11692.

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing Properties of Neural Networks, In: International Conference on Learning Representations, 2014.

[20] A. Aghajanyan, A. Shrivastava, A. Gupta, N. Goyal, L. Zettlemoyer, S. Gupta, Better Fine-ti, Arxiv Preprint Arxiv:2008.03156.

[21] E. Loper, S. Bird, Nltk: The Natural Language Toolkit, Corr Cs.Cl/0205028.

[22] D. Q. Nguyen, D. Q. Nguyen, T. Vu, M. Dras, M. Johnson, A Fast and Accurate VietnameseWord Segmenter, In: Proceedings of The Eleventh International Conference on Language Resources and Evaluation (Lrec 2018), European Language Resources Association (Elra), Miyazaki, Japan, 2018.

[23] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, M. Johnson, Vncorenlp: A Vietnamese Natural Language Processing Toolkit, In: Proceedings of The 2018 Conference of The North American Chapter of The Association for Computational Linguistics: Demonstrations, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 56–60.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. Devito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An Imperative Style, High-Performance Deep Learning Library, In: H. Wallach, H. Larochelle, A. Beygelzimer, F. D'alché-Buc, E. Fox, R. Garnett (Eds.), Advances In Neural Information Processing Systems, Vol. 32, 2019.

[25] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. Von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-Of-The-Art Natural Language Processing, In: Proceedings of The 2020 Conference on Empirical Methods In Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2020, pp. 38–45.

[26] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, In: Y. Bengio, Y. Lecun (Eds.), 3rd International Conference on Learning Representations, Conference Track Proceedings, 2015.

[27] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised Cross-Lingual Representation Learning at Scale, In: Proceedings of The 58th Annual Meeting of The Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 8440–8451.