An Empirical Study of Vietnamese Machine Reading Comprehension with Unsupervised Context Selector and Adversarial Learning

Hoang Vu Tran and Phuc Minh Nguyen

November 22, 2021

An empirical study of Vietnamese Machine Reading Comprehension with Unsupervised Context Selector and Adversarial Learning

Vu Hoang Tran
Vinbrain
v.vutran@vinbrain.net

Minh Phuc Nguyen
Vinbrain
v.minhng@vinbrain.net

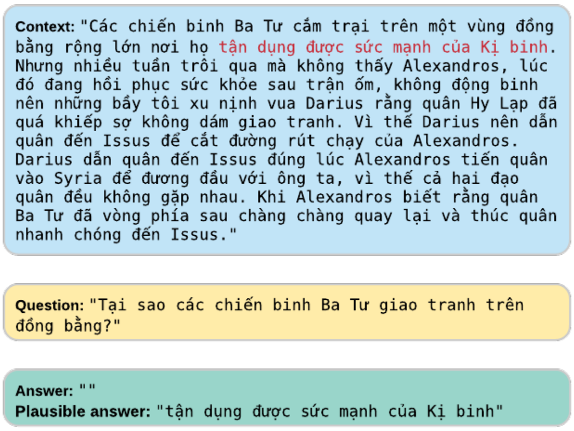
Abstract

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 lots of 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-MRC shared task (Nguyen et al., 2021) 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.

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 dataset (Nguyen et al., 2021).

PTMs that can capture contextual word embeddings such as ELMo (Peters et al., 2018), GPT (Cohen and Gokaslan, 2020), or BERT (Devlin et al., 2019) have proposed and achieved superior results. But all these PTMs only experiments on English language (Hermann et al., 2015; Lai et al., 2017; Rajpurkar et al., 2016, 2018; Nguyen et al., 2016).



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-MRC shared task dataset. The highlighted span text in context is the plausible answer for the question.

The most well-known pre-trained models, such as BERT (Devlin et al., 2019), are used on fixed-length input segments of a maximum of 512/1024 tokens owing to the limitation of fixed-length. Thus, a long input must be partitioned into smaller segments of manageable sizes. It leads to the loss of salient cross-segment information, that is, the context fragmentation problem. (Dai et al., 2019; Ding et al., 2021) proposed new architecture to solve this problem.

Adversarial training (AT) (Goodfellow et al., 2015) is a means of regularizing classification algorithms by generating adversarial noise to the training data. In the Machine Reading Comprehension task, (Lee et al., 2019) leveraged AT for learning domain-invariant representation, which made the MRC model generalize well to predict answers on unseen out-of-domain. (Yang et al., 2019) applied Vitural Adversarial Training to improve the performance significantly and universally on SQuAD1.1 (Rajpurkar et al., 2016), SQuAD2.0 (Rajpurkar et al., 2018) and RACE. (Yang et al., 2021) propose a novel adversarial training method called PQAT. The core of PQAT was the virtual P/Q-embeddings, which were two independent embedding spaces for passages and questions. According to the benefits of AT, we decided to apply severals training strategies that can boost the model performance across in MRC tasks which is discussed further in Section 2.2 and Section 2.3.

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-MRC shared task (Nguyen et al., 2021).

2 Background

2.1 Pre-trained Language Models

PTMs on the large unlabeled corpus have shown impressive performance on lots of downstream NLP tasks which proves 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 (Peters et al., 2018), GPT (Cohen and Gokaslan, 2020), or BERT (Devlin et al., 2019) to transfer the knowledege 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 (Wang et al., 2018). In terms of monolingual language models pre-trained for Vietnamese, PhoBERT has been introduced by (Nguyen and Nguyen, 2020) and it has shown significant improvements in Named Entity Recognition, Parsing, and Natural Language Inference tasks. PhoBERT pre-training approach is based on RoBERTa (Liu et al., 2019) 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 enable 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 concanation of C and Q with special tokens $[CLS]$ and $[SEP]$ as $[CLS]$ Q $[SEP]$ P $[SEP]$. We employ a linear layer with Softmax operation and feed last-layer hidden representation $H \in \mathbb{R}^{LXd}$ as the input to obtain the start/end position probability distributions p_s, p_e respectively. The training objective of answer span prediction is defined as cross entropy loss for the start and end index position.

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

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 $h_{CLS} \in H$ and use cross entropy as loss function for classification answerable/unanswerable question.

$$loss_{CLS} = -\frac{1}{N_k}\sum_{k}^{N_k}\sum_{c}^{C}[y_c^k log(p_c^k)] \tag{2}$$

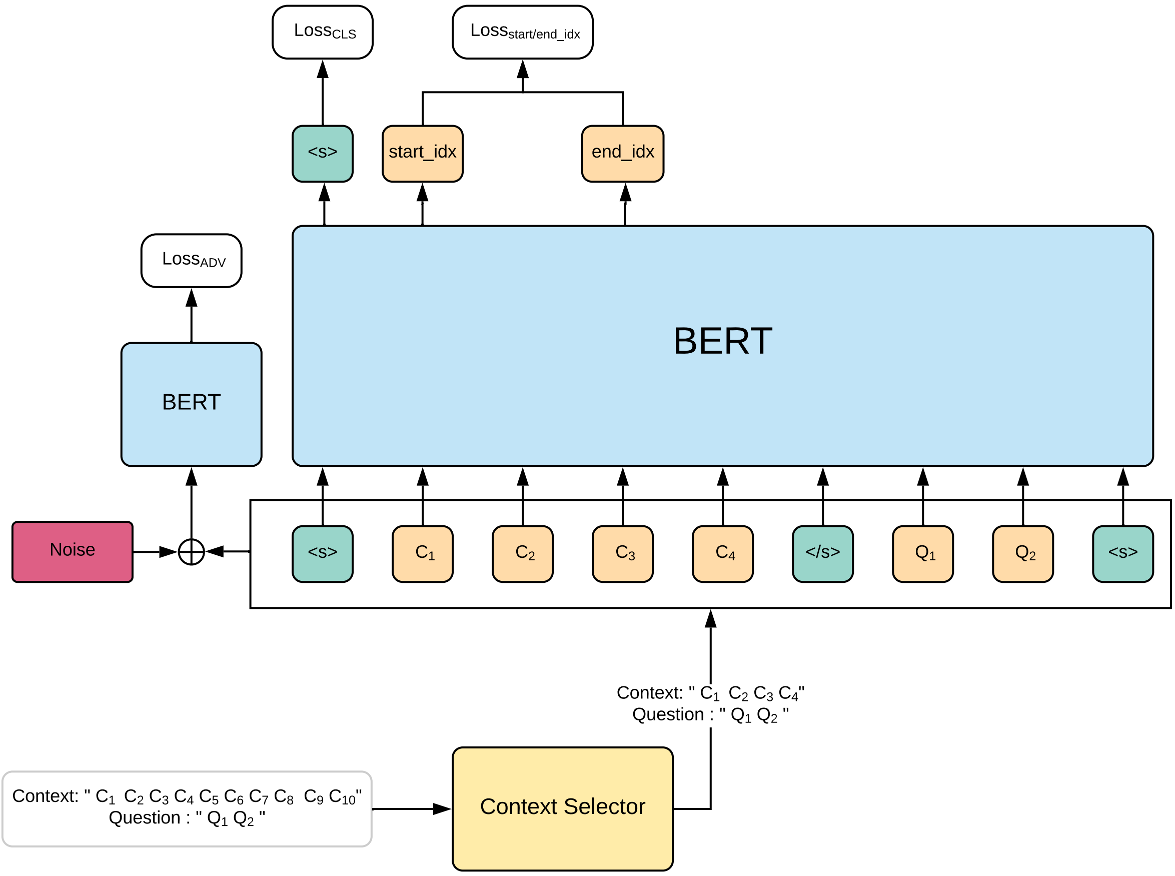
where p_c^k is answerable and unanswerable probability distributions. C means the number of classes (C = 2 in this work).

The overview of our method architecture is illustrated in Figure 2 .

2.2 Adversarial Training

(Szegedy et al., 2014) first discovered the existence of small perturbations to the input images that mislead models to predict wrong labels in the image classification. They called the perturbed inputs adversarial examples. (Goodfellow et al., 2015) proposed a simple adversarial training method to improve the robustness of the model by training on both clean examples and adversarial examples. 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:

$$y = f_\theta(x) \tag{3}$$



Loss$_{CLS}$
Loss$_{start/end\_idx}$
<s>
start_idx
end_idx
Loss$_{ADV}$
BERT
BERT
Noise
<s>
C$_1$
C$_2$
C$_3$
C$_4$
</s>
Q$_1$
Q$_2$
<s>
Context: " C$_1$ C$_2$ C$_3$ C$_4$"
Question : " Q$_1$ Q$_2$ "
Context: " C$_1$ C$_2$ C$_3$ C$_4$ C$_5$ C$_6$ C$_7$ C$_8$ C$_9$ C$_{10}$"
Question : " Q$_1$ Q$_2$ "
Context Selector

Figure 2: The overview architecture of our method.

$$y' = f_\theta(x + noise) \tag{4}$$

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 simply is a noise layer for the input. In this work, we utilize R3F (Aghajanyan et al., 2020) that encourage the model to generalize with representation changes during training without hurting performance. The adversarial training loss $loss_{ADV}$ is calculated by the following:

$$loss_{ADV} = KL(y, y') + KL(y', y) \tag{5}$$

where KL is the KL-Divergence. The final loss function is the summation of mentioned loss with λ_0, λ_1 and λ_2 are learned weight for each task:

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

2.3 Domain Agnostic (DA)

Adapting models to a new domain without finetuning 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 sample of class k and N is total samples.

3 Method

3.1 Unsupervised Context Selector

Due to the input sequence may exceed the beneficial length of BERT (Devlin et al., 2019) (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 given 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 [1]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 (Loper and Bird, 2002), 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 score $score(s) > 2\epsilon$ following:

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

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

where $\epsilon = \sum_{t \in s; score(t) \neq 0} score(t)$, $g(t)$ 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 2, there is an imbalance between answerable and unanswerable questions. This makes the model easily predict plausible answers and mistaken the fact of the given context and question. We introduce a simple strategy to generate unanswerable examples from the training set, called Question-Context shuffle. This approach aims to augment more unanswerable samples by for each given context, we get a random irrelevance question.

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

Table 1: Statistic of classes of training dataset after data augmentation.

We divide the generated unanswerable samples into two types are EXAMPLES$_{hard}$ and EXAMPLES$_{easy}$. The EXAMPLES$_{hard}$ are examples where the question and the passage are in the same title but different contexts. Otherwise, the examples that have different titles are categorized into EXAMPLES$_{easy}$. The statistic of the dataset after pre-processing is presented in Table 1 in which the total samples of two class has been balanced.

4 Experimental Results

4.1 Set up

We employ RDRSegmenter (Nguyen et al., 2018) from VnCoreNLP (Vu et al., 2018) to perform word-level and sentence segmentation on UIT-ViQuAD dataset (e.g "Những cá_thể xung_quanh ghi_nhớ tôm tít bằng cách nào ?"). Our experimental models were implemented PyTorch (Paszke et al., 2019) and utilize Huggingface's Transformers (Wolf et al., 2020) 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 [2]training. In the first phase, we make the model generalize with and warm up with the data by setting the $\lambda_0 = 0.2$, $\lambda_1 = 0.6$, and $\lambda_2 = 0.2$. We observed that the $loss_{ADV}$ is converged after the first phase, we decided to set $\lambda_2 = 0$ on every next phase. In the second phase, we aim to make the classification loss which only saves the checkpoint that has 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

[1] https://github.com/undertheseanlp/underthesea

[2] We also explored classifying answerable questions and predicting answer spans as two separated modules before train end-to-end these models, but did not observe any improvements.

	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

Table 2: 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.

on dev set. We set the $\lambda_0 = 0.9$ and $\lambda_1 = 0.1$ on the second phase and $\lambda_0 = 0.1$ and $\lambda_1 = 0.9$ on the third phase.

4.2 Dataset

In VLSP2021-MRC shared task (Nguyen et al., 2021), 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 2. Since there is not any dev set, we decided to categorize the articles in the training dataset into two main sets based on answerable and unanswerable questions which make the split dataset is 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 (Kingma and Ba, 2015) 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. For each sample, we set the maximum sequence length for context and questions to be 230 and 50 respectively. All experiments are launched with a maximum of 10 epochs and single A100-40GB GPU device.

4.4 Results

4.4.1 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 (Nguyen and Nguyen, 2020)) perform better than multilingual models (e.g mBERT (Devlin et al., 2019), XLM-Roberta (Conneau et al., 2020)). 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 Tabel 3.

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

	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

Table 4: Results on Vi-SQuAD private test set. $\star$ refer to word-level.

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 4.5. The result of the top-3 team and our result in the private test is illustrated in Table 4.

4.4.2 Context Selector

We also evaluate our unsupervised Context Selector on the train set which is shown in Table 5. 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

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

Table 5: Results of Context Selector on Vi-SQuAD train set. * refers samples that has context length > 256 syllable tokens

Context Selector has crucially reduced the context length while retaining the answer in the filtered context.

4.5 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 6 shows a few span error examples that we have analyzed in VLSP2021-MRC shared task.

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 are 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 the ability to handle multi-hop questions is limited. Moreover, the surrounding context of the answer may no sufficient or related to the answer in the filtered context which may hurt the model on prediction.

5 Conclusion

We introduce applied Context Selector to overcome the large context problem, which is a prominent 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-MRC shared task 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%.

Question:"Lịch sử của Ba Tư được ghi chép vào năm nào?" **Label**:"khoảng năm 3200 TCN" **Pred**:"năm 3200 TCN"
Question:"Hiện tại, một cuộc tranh cãi đang nổ ra về vấn đề nào" **Label**:"nguồn gốc các tên gọi của thực thể - Iran và Persia" **Pred**:"nguồn gốc các tên gọi của thực thể - Iran và Persia (Ba Tư)"
Question:"Mâu thuẫn giữa Iran và Mỹ ngày càng leo thang ở vấn đề nào?" **Label**:"chương trình hạt nhân của Iran" **Pred**:"Vấn đề chương trình hạt nhân của Iran"

Table 6: 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.

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2020. Better fine-tuning by reducing representational collapse. *arXiv preprint arXiv:2008.03156*.

Vanya Cohen and Aaron Gokaslan. 2020. Opengpt-2: Open language models and implications of generated text. *XRDS*, 27(1):26–30.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

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

SiYu Ding, Junyuan Shang, Shuohuan Wang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021.

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 (Volume 1: Long Papers)*, pages 2914–2927, Online. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Seanie Lee, Donggyu Kim, and Jangwon Park. 2019. Domain-agnostic question-answering with adversarial training. In *EMNLP*.

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

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028.

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. PhoBERT: Pre-trained language models for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1037–1042.

Dat Quoc Nguyen, Dai Quoc Nguyen, Thanh Vu, Mark Dras, and Mark Johnson. 2018. A fast and accurate Vietnamese word segmenter. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kiet Van Nguyen, Son Quoc Tran, Luan Thanh Nguyen, Tin Van Huynh, Son T. Luu, and Ngan Luu-Thuy Nguyen. 2021. Vlsp 2021 shared task: Vietnamese machine reading comprehension. In *Proceedings of the 8th International Workshop on Vietnamese Language and Speech Processing (VLSP 2021)*.

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. *CoRR*, abs/1611.09268.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2018. 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*, pages 56–60, New Orleans, Louisiana. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. 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*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le

Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ziqing Yang, Yiming Cui, Wanxiang Che, Ting Liu, Shijin Wang, and Guoping Hu. 2019. Improving machine reading comprehension via adversarial training. *ArXiv*, abs/1911.03614.

Ziqing Yang, Yiming Cui, Chenglei Si, Wanxiang Che, Ting Liu, Shijin Wang, and Guoping Hu. 2021. Adversarial training for machine reading comprehension with virtual embeddings. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 308–313, Online. Association for Computational Linguistics.