ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Machine Reading Comprehension Using Multi-Passage BERT with Dice Loss on Thai Corpus

Theerit Lapchaicharoenkit[1] and Peerapon Vateekul[2]

ABSTRACT

Nowadays there is an advancement in the field of machine reading comprehension task (MRC) due to the invention of large scale pre-trained language models, such as BERT. However, the performance is still limited when the context is long and contains many passages. BERT can only embed a part of the whole passage equal to the input size. Thus, sliding windows must be used, which leads to discontinued information when the passage is long. In this paper, we propose a BERT-based MRC framework tailored for a long passage context on a Thai corpus. Our framework employs the multi-passage BERT along with self-adjusting dice loss, which can help the model focus more on the answer region of the context passage. We also show that there is an improvement in the performance when an auxiliary task is used. The experiment was conducted on the Thai Question Answering (QA) dataset used in the Thailand National Software Competition. The results show that our method improves the model's performance over a traditional BERT framework from 0.7614 to 0.7742 in terms of F1, especially on longer passages with more 2,000 tokens and more.

Article information:

Keywords: Machine Reading Comprehension, Natural Language Processing, Deep Learning, BERT

Article history:
Received: May 22, 2021
Revised: October 18, 2021
Accepted: March 4, 2022
Published: May 28, 2022
(Online)

DOI: 10.37936/ecti-cit.2022162.247799

1. INTRODUCTION

In recent years, we have seen a surge in the availability and popularity of MRC research and datasets. Such research creates a machine that possesses the ability to read and comprehend a piece of document and then is able to answer questions related to the document. There are several types of subproblems in MRC and different datasets have been created having different objectives and motivations. SQuAD (Stanford Question Answering Dataset) [1] is a notable example of a public, large-scale English MRC datasets, composed of more than 100,000 extractive question-answer pairs. Natural Questions [2] is similar to SQuAD. It also utilizes Wikipedia articles as context passages. Natural Questions tasks the model to select both answers and paragraphs that contain answers. TriviaQA [3] and SearchQA [4] are examples of datasets that deal with the problem of Open-Domain QA where context passages are not paired or matched with the questions. Models must query the context or document passage themselves. This line of research reflects the application more.

Besides the English language, large-scale MRC datasets in other languages also exist, such as DuReader [5], which is a Chinese MRC dataset. Although there are many QA datasets in English, Thai QA datasets are more scarce. An example of such a QA dataset is [6], which is a Thai language dataset from Thailand's 22nd National Software Competition (NSC). Apart from other issues involving Thai natural language processing (NLP) tasks, one of the biggest challenges in this dataset is that each context can span multiple passages. This can decrease the model's performance since the computational cost is higher and there are more answer candidates.

In earlier MRC research, models consisting of a series of RNN variants and attention layers are normally used. Examples of this model include BiDAF (Bi-directional Attention Flow) [7], and FusionNet [8]. In 2018, an advanced pre-trained language model called BERT [9] was released. BERT (Bidirectional Encoder Representations from Transformers) is a large, deep learning model consisting of multiple layers of transformer architecture [10] and is pre-trained

[1,2] The authors are with Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand, E-mail: 6170932121@alumni.chula.ac.th and peerapon.v@chula.ac.th
[2] Corresponding author: peerapon.v@chula.ac.th

on a large amount of unlabeled corpus. BERT can be applied to a variety of natural language processing tasks including sentiment analysis, natural language inferences, and MRC, and can achieve good results. After the introduction of BERT, newer MRC research focuses on the utilization of a large pre-trained language model (LM) through task-specific fine-tuning.

Herein, our scope of the study is classified as extractive, multi-passage MRC. Similar to the extractive MRC problem, the MRC model finds the answer to questions by predicting the starting and ending positions of the answers which can be found in the context of the passage. The difference from normal MRC is that the length of the context of the passages can span multiple paragraphs. This problem is challenging as it takes more time to train the model, and there is a chance that most of the text that the model has to read may contain information that is not useful in answering questions. Fine-tuning BERT or other pre-trained language models usually achieves strong performance when context information is contained in one paragraph or the length of the context passage is shorter than the sequence input size of the pre-trained LMs. However, if the length of the passage is longer than the model's input sequence length, the performance of the model can deteriorate as the context of the passage must be split into multiple windows. When the context passages are longer than the pre-trained LM sequence input size, sliding windows are used to consecutively feed the context and question into the model. The probabilities of starting and ending positions for these windows are computed separately, so the scores may not be directly comparable [11].

In this paper, the aim is to create a multi-passage MRC framework for a Thai corpus. The multi-passage BERT technique, therefore, is integrated into our framework to tackle the questions contained in the long passages. Multi-passage BERT uses the concept of global normalization across different context passage windows. This makes the computation of starting and ending position probabilities more comparable across different windows. This concept is further extended via the usage of self-adjusting dice loss [12], which helps the model pay more attention to the area of answer positions and predicted positions. This loss is more favorable to the traditional loss (cross-entropy loss) that takes all tokens into consideration, which may not be suitable for the multi-passage setting. Similar to the original multi-passage BERT work, a modified auxiliary task is also implemented.

To summarize, the contribution of our work is listed below:

- An MRC model is presented, which is based on multi-passage BERT, designed specifically to handle long context passages, commonly found in Thai QA data sets.
- Self-adjusting dice loss is employed rather than a traditional loss (negative log-likelihood) since it is more suitable for long passages where the majority of the tokens are negative examples.
- An auxiliary task is proposed and added to our network to further improve accuracy. In a long passage, the input must be divided into several windows. An auxiliary classifier can help to locate whether or not an answer is in each window.

In Section 2, the related works are discussed. In Section 3, details of our methodology are discussed. Section 4 deals with details of the dataset and the implementation of our experiment. Section 5 analyzes the results and contains a discussion of the experiment. The conclusion is found in Section 6.

2. RELATED WORKS

This section aims discusses the techniques in the domain of machine reading comprehension (MRC), especially for long passages (multiple passages). There are two main parts, which include a traditional technique (non-BERT) and a BERT-based technique.

2.1 Traditional MRC Techniques

Many works have been conducted which performs MRC tasks in long context passages or multi-passage MRC settings. Memoreader [13] has proposed a model that can deal with long-range dependency in MRC tasks through the use of memory controlling units and encoding blocks, which utilize GRU and self-attention. Wang et al. [14] have also conducted research into the multi-passage aspects of MRC. In addition, in order to span prediction tasks, the authors have included auxiliary tasks in the model and employ cross-passage answer verification through the use of an attention mechanism. After that, Wang et al. [15] proposed employing the use of reader and ranker architecture coupling it with reinforcement learning for the multi-passage MRC tasks. HasQA [16] proposed a hierarchical framework approach for dealing with the Open-Domain QA tasks, where answers are the product of the probability of a paragraph containing the answer or not, and conditional answer probability, which reflects the quality of a different answer span candidates for a given paragraph. However, all of these works are not based on pre-trained LM like BERT [9], so they do not have the benefit of pre-training on a large unlabeled corpus.

In Thai MRC, most prior research has been based on traditional MRC techniques, not BERT. Jitkrittum et al. [17] developed a Thai QA system on data collected from Wikipedia and implemented a rule-based approach to match keywords to extract answers. Kongthon et al. [18] tackled Thai QA tailored for the tourism domain using an information retrieval approach, converting questions to a unified query and then using it to search for an answer in the database. Recently, Noraset et al. [19] carried out an extractive approach for Thai QA using bidirectional Long-Short-Term-Memory (LSTM) [20] coupled with an attention mechanism. Such a rule-based approach needs humans to manually create predefined rules, which can be used for a specific domain. For the deep learning approach, BERT has been proven to outperform other recurrent neural networks such as LSTM and Bi-LSTM, especially for QA tasks [9], as well as on similar MRC datasets in English (SQuAD).

2.2 BERT-Based MRC Techniques

BERT [9] is a pre-trained LM built upon a transformer architecture [10]. A transformer is a deep learning network comprised solely of an attention mechanism that differs from prior NLP works where LSTM or bidirectional LSTM are used to process information before an attentional layer is applied. Using a transformer leads to better computational speed and performance in machine translation tasks compared to recurrent-based models [10]. Transformer modules consist of self-attention and feed-forward layers, layer normalization [21], and residual connections [22], as shown in Fig. 1. [23]

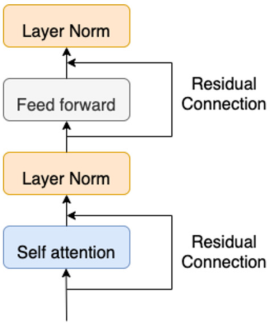

Fig.1: Transformer module, consisting of different deep learning network layers.

BERT is made of several transformer modules and is designed to be a model that can be generally adapted to fulfill various NLP tasks. BERT is thus pre-trained on a large and unlabelled corpus using the language modeling task. After the pre-training process, BERT can be readily fine-tuned to various NLP tasks such as MRC, semantic classification, and natural language inference. The number of transformer modules can be varied in different versions of BERT. In our study, we use normal BERT, which is made of 12 transformer modules and is illustrated in Fig. 2.

Similar to applications of BERT on other NLP tasks [9], BERT for MRC works by introducing an additional fully connected neural network layer, or a fine-tuning layer to the architecture. In BERT MRC, these dense layers process the representation vectors produced by BERT and predict the probabilities of certain tokens being the starting positions or ending positions of the answer. The architecture of BERT MRC is given in Fig. 3. In subsequent illustrations of BERT architecture, the model architecture is simplified down to a simple box representing one window. During training, the negative log-likelihood is used to calculate the loss.

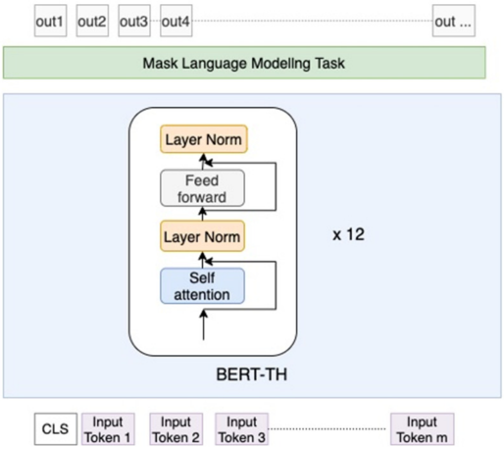

Fig.2: General architecture of BERT.

For questions with multiple passage paragraphs, the sliding window approach is normally used [11]. In this setting, multiple pairs of question-context are created and fed into BERT to produce the starting and ending positions of the answers. In the training phase, each window has a pair of labeled starting and ending positions, as shown in Fig. 4. If a window does not contain an answer, the starting and ending positions are changed to the position of the CLS token instead.

During inference time, the window with the highest probability scores will be used as the final prediction of the question. It must be pointed out that that the output from each BERT window is processed separately and information from other or nearby windows is not used, so the scores from each window may not be directly comparable [11], [24]. Multi-passage BERT research tackles this issue.

Multi-passage BERT [11] is a technique that deals with MRC tasks on Open-domain QA datasets, where answers are normally located in multi-passage documents. In their work, BERT MRC was employed by modifying the output vector normalization process and using a passage ranking module. Additionally, information retrieval by BM25 was used to select top passages for the MRC framework. In this paper, our model is based on Multi-passage BERT with several custom modifications. These modifications are discussed in the following section.

3. METHODOLOGY

In this section, details of our algorithm based on the multi-passage BERT are provided. It is specifically designed for Thai MRC having a long passage context. There are three main modifications: (1) self-

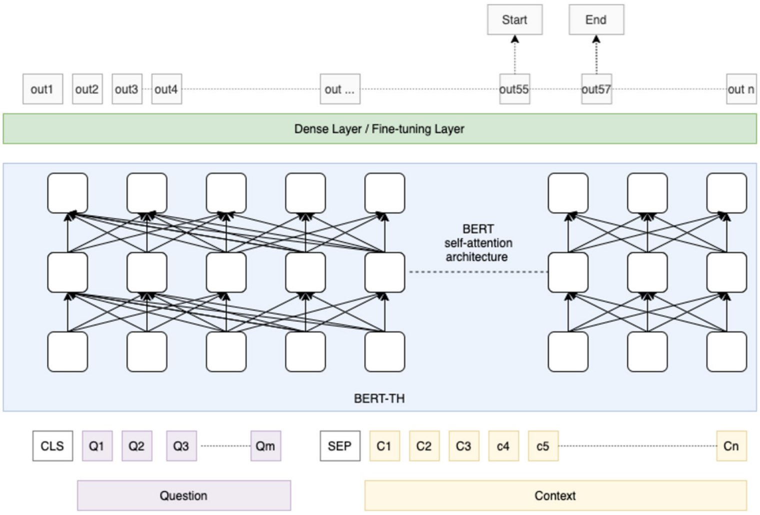

Fig.3: *The architecture of BERT MRC. Question and context data are fed together into the model, separated by special tokens.*

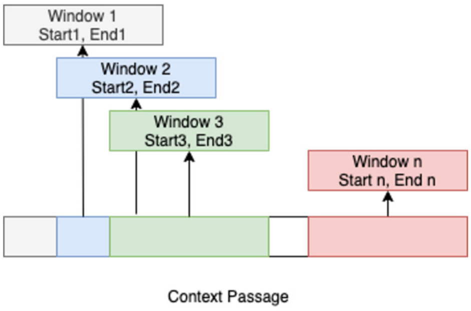

Fig.4: *A long context passage is divided into several fixed-size, overlapping windows. The output of each window is a pair of starting and ending positions.*

adjusting dice loss, (2) multi-passage score normalization, and (3) an auxiliary task.

3.1 Self-Adjusting Dice Loss

Li, et al. [12] conducted research on the application of dice loss in NLP. The authors tested several variations of dice loss and proposed self-adjusting dice loss. This self-adjusting dice loss was designed to be used in tasks that have a large number of easy negative examples. These can be found in MRC problems where most of the tokens are negative examples (non-answer positions), while there is only one starting or ending position. Self-adjusting dice loss works by taking only the answer and predicted tokens into account, rather than taking tokens from all context passages. For a long context passage, it is not suitable to use traditional loss since the number of non-target words is very large and can affect the performance of the model. Thus, it is more appropriate to calculate loss around the answer locations.

Self-adjusting dice loss is derived from the F1 score, which is shown in Eq.(2) below. The loss is calculated based on the intersection areas of the text that are actual starting and ending positions (positive class) and predicted positive classes only, rather than calculating the loss from all positions of the texts, as shown in Eq (1):.

$$\begin{aligned} loss_{CE} = -y_{pos} * \log(P_{pos}) - (1 - y_{pos}) \\ * \log(1 - P_{pos}) \end{aligned} \tag{1}$$

$$loss_{DSC} = 1 - \frac{2 * (1 - P_{pos}) * P_{pos} * y_{pos} + \gamma}{(1 - P_{pos}) * P_{pos} + y_{pos} + \gamma} \tag{2}$$

3.2 Multi-Passage Score Calculation

In multi-passage BERT, the logit outputs, which are from all BERT windows, are globally normalized. This contrasts with single-passage BERT MRC implementations where logit scores are normalized across the values from the same BERT window only, as shown in Fig. 5a. The process of normalizing the scores across all passages from the same question is illustrated in Fig. 5b. The target of loss calculation in multi-passage BERT is similar to BERT MRC. For example, if there are 10 windows for a question, there will be 10 starting and ending positions for loss calculation for that question.

Wang, et al. [11] experimented with different setting configurations for multi-passage BERT, ranging from using non-overlapping windows to the usage of passage ranking. Similarly, it has also been found that using multi-passage BERT with overlapping windows leads to the best results in our preliminary experiments.

Using the self-adjusting dice loss discussed in the previous section allows us to obtain better score calculation. Instead of approaching the loss calculation by averaging the loss across different windows, we concatenate all the window passages together and pick only one starting and ending position as ground truth labels, as illustrated in Fig. 5c. This approach allows us to fully utilize the self-adjusting dice loss by calculating the loss from the unified passage rather than calculating the loss from separate, smaller windows.

Since the sliding window approach is used for multi-passage BERT [11], starting and ending answer positions may appear more than one time in the concatenated representations. We mark windows that have starting and ending positions in the center as final ground truth labels. Eqs. (3) and (4) describe that concatenation. This approach of combining multiple windows is similar to [25]. It is evident that this label formulation, coupled with self-adjusting dice loss [12], increases the model's performance.

$$P_{start} = Softmax([W \times h_{x1} : W \times h_{x2} :, \ldots]) \quad (3)$$

$$P_{end} = Softmax([W \times h_{x1} : W \times h_{x2} :, \ldots]) \quad (4)$$

where h_{xn} represents the output representation from BERT at the n^{th} window. W is the collection of trainable parameters of the fully connected layer. In our method, all hidden representations are concatenated together before normalization is carried out and before applying the loss calculation.

3.3 Auxiliary Classification Task

To improve the performance of the model, it is common to incorporate an auxiliary task as in the original multi-passage BERT. In this work, there is a slight modification. For the auxiliary task, we propose using classification to decide if a window contains an answer or not without ranking the passage. Additionally, we share the model parameters for the classification task with the model parameters for the answer prediction task. Eqs. (5) to (7) describe our usage of the classification module:

$$P_{cls} = W \times h_{CLS} \quad (5)$$

$$P_{start} = P_{start} \times P_{passage} \quad (6)$$

$$P_{end} = P_{end} \times P_{passage} \quad (7)$$

where h_{CLS} represents the BERT output representation of the special token [CLS], which is normally used in the passage classification task. W is collection of trainable parameters of the dense layer. $P_{passage}$ is the probability of a certain passage containing the ground truth answer. The classification score of each passage is normalized across all context passages having the same question.

Our complete model can be seen in Fig. 6. This model contains 3 modifications, which include the usage of self-adjusting dice loss, a new score calculation method that uses only one starting and ending position, and usage of the classification task.

4. EXPERIMENTS

This section aims to describe the experimental dataset and its statistics, the implementation details of our model which includes the hyperparameters, and model performance evaluation.

4.1 Dataset

In this paper, the experiment was conducted on the NSC QA dataset using only factoid questions that have been matched with the context passages already. Table 1 shows data statistics for all 15,000 factoid questions. It is evident that the context is very long (containing multiple passages) and must be divided into multiple windows. However, the answers are quite short, so only a few windows contain an answer (starting and ending positions).

Table 1: *Dataset statistics of Thai NSC factoid questions. It shows the number of tokens (length) in each part including questions, context, and answers.*

#Tokens in Each Part	Min	Mean	Max
Question	3	14	52
Context	11	735	20,749
Answer span	1	1.7	33

Table 2 shows an example of factoid questions in the NSC QA dataset. Additionally, Fig. 7 shows a histogram of the number of passages. There are around 1,300 out of 15,000 examples that are found in one passage. Since the context is provided in one continuous plain text paragraph, a passage is defined as a set of 100 words.

4.2 Implementation Details

For Thai language text processing, before putting text into the BERT model, a newmm tokenizer from PythaiNLP is used [26]. The Thai tokenization process is applied for both questions and context alike. For the model, the hyperparameters for both BERT MRC [9] and multi-passage BERT [11] are as follows: 1) for the model's weight inside the BERT architecture, the Thai version of the model is used, 2) Adam [27] is utilized as an optimizer, the learning rate is set at 3*10-5 and the epsilon is set to 1*10-8, which are the default values for BERT MRC implementation, and 3) the dense classifier hidden dimension is 50. Using a max sequence length of 128 and a stride length of 65 provides the best results for both BERT MRC and multi-passage. The model's checkpoint selection bases its performance on the validation set.

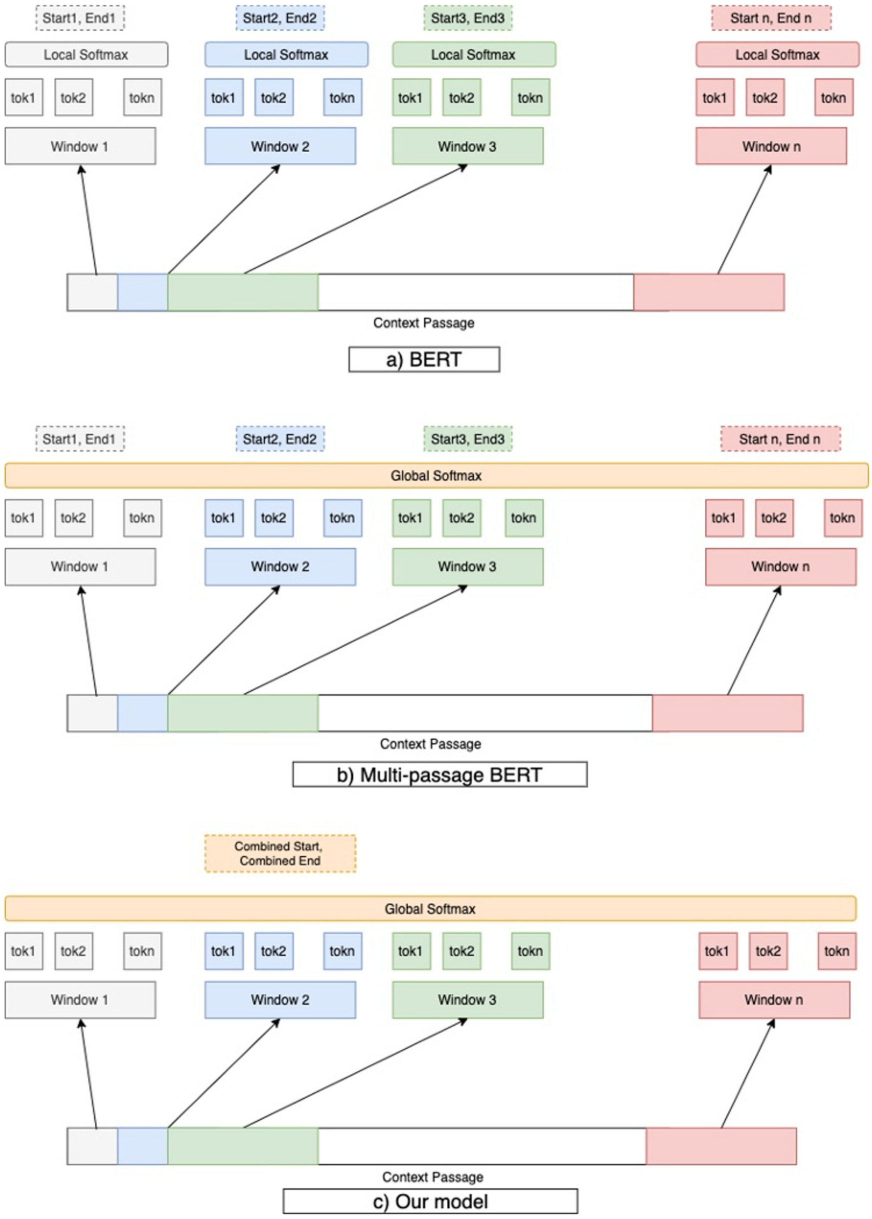

Fig.5: *Starting and ending positions used in score calculation in the multi-passage context in different models: (a) BERT, (b) multi-passage BERT, and (c) our model*

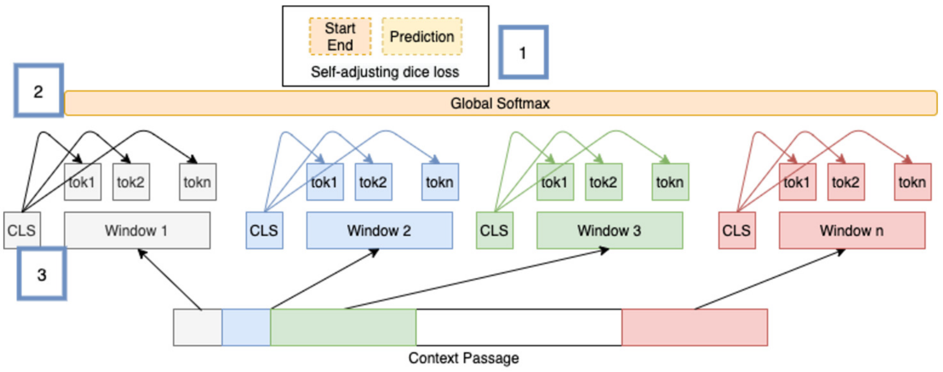

Fig.6: *Our full model that utilizes the combined window score calculation and classification task.*

In terms of the multi-passage windows, the maximum number of windows used during the training is 20 due to memory limitations, and the batch size is set to 1. The batch size for BERT MRC is set to 24. For the self-adjusting dice loss implementation, gamma is set at 1, similar to the original paper [12].

Table 2: *Examples of Thai factoid questions in the NSC QA dataset.*

Question (#tokens = 33)
Thai: พีต้า เมลลาร์ก ตัวละครหลักในเรื่อง "ไตรภาคเกมล่าชีวิต" เป็นตัวแทนในฐานะบรรณาการหลักจากเขตที่เท่าใดของประเทศพาเน็ม English: Peeta Mellark, the main character of the Hun ger Games trilogy, is the representative tribute of whi ch district?
Context (actual #tokens = 548)
Thai version: พีต้า เมลลาร์ก เป็นตัวละครหลักจากไตรภาคเกมล่าชีวิต ของซูซาน คอลลินส์ เขาเป็นลูกชายของคนขายขนมปังอาศัยอยู่ในเขต 12 ของประเทศพาเน็ม ในหนังสือเล่มแรก พีต้าต้องเข้าแข่งขันเกมล่าชีวิตในฐานะบรรณาการจากเขต 12 English version: Peeta Mellark is the main character of the Hunger Games trilogy by Suzanne Collins. He is the son of the bakers who live inside district 12 of Panem. In the first book, Peeta must compete in the hunger game as a tribute from district 12.
Answer
Thai: เขต 12 English: District 12

Table 3: *Token-level F1 score calculation.*

Predicted Tokens	-	ผู้	หญิง
Actual Tokens	เด็ก	ผู้	หญิง
Precision	0.333		
Recall	0.667		
F1	0.444		
EM	0		

4.3 Evaluation Metrics

We use 2 main evaluation metrics normally used in extractive MRC tasks, which are token-level F1 and Exact Match (EM). Token-level F1 measures the partial overlap between the predicted answers and ground truth answers. While EM score checks for a complete match between the prediction and ground truths. We report the results of our experiments based on 3-fold cross-validation splits.

For token-level F1 calculations, the predicted tokens, as well as the actual (ground truth) tokens, must be calculated from precision and recall. This is shown in Table 3 and Eqs. (8) to (10). The newmm tokenizer from PythaiNLP is used [26].

$$Precision = \frac{\#\ correctly\ predicted\ tokens}{\#\ total\ predicted\ tokens} \tag{8}$$

$$Precision = \frac{\#\ correctly\ predicted\ tokens}{\#\ total\ actual\ tokens} \tag{9}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

5. RESULTS AND DISCUSSION

From Table 4, results demonstrate that our model indeed improves the performance from the baseline, BERT (original), with 0.6287 exact match (EM) and 0.7742 of F1-score. This aligns with the intuition that the self-adjusting dice loss and the auxiliary task are suitable for a question having a long passage context.

Additionally, Table 4 shows that the multi-passage BERT (2nd row) surprisingly performs worse than the original BERT (1st row), in which F1 scores are 0.7375 and 0.7614 respectively. Such a poor outcome is due to the difference in batch sizes between the two models. Since BERT MRC normalizes the probabilities separately, each batch that is passed to the graphic processing unit (GPU) consists of examples from different questions. On the other hand, each batch that is fed into the multi-passage BERT model must be comprised of examples from the same questions. This may lead to better regularization and make BERT MRC perform better than the multi-passage BERT.

It is significant to note that multi-passage BERT is still more suitable than BERT in many aspects and shows better results when combined with other strategies. Integration of the self-adjusting dice loss helps increase the multi-passage BERT performance as expected, increasing the performance from 0.7614 to 0.7728 in terms of F1, and 0.6160 to 0.6296 in terms of EM.

Another point to note is that similarly to the original multi-passage BERT, the incorporation of auxiliary task scores also gives better model results than variants without the auxiliary classification tasks. This shows that the auxiliary tasks are effective in helping the model focus on the correct or relevant window by assigning probability scores. Results ultimately led to an increase in F1 from 0.7728 to 0.7742.

In Table 5, the effectiveness of our method in terms of performance by context passage length is illustrated. The context of the passages is split into different groups based on the number of tokens that the context passages had. Note that integration of

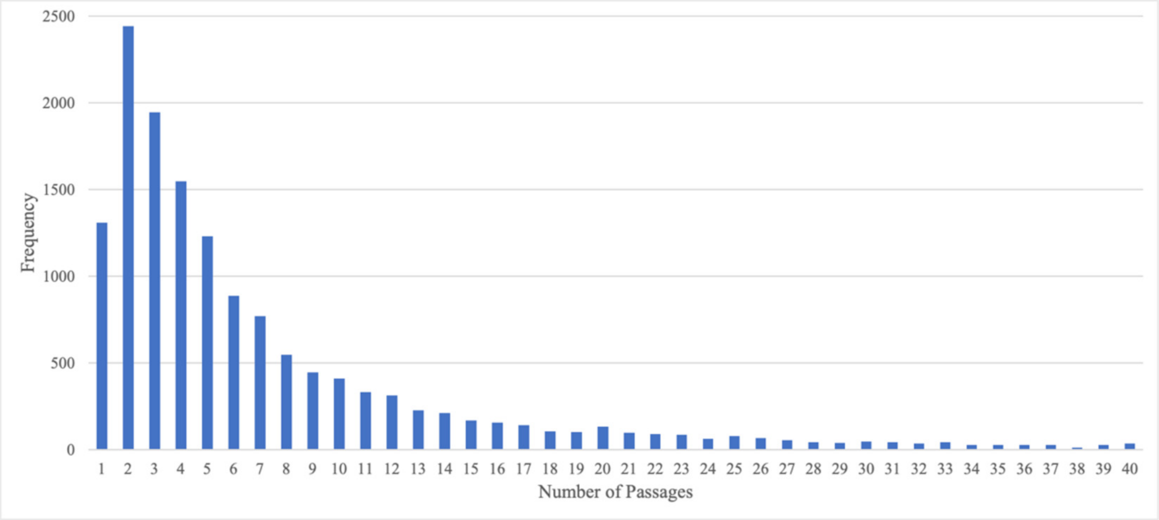

Fig. 7: *Distribution of the number of passages in the context.*

Table 4: *Results of the experiments. Values in bold faces have the best performance. (DL denotes self - adjusting dice-loss, Aux denotes Auxiliary task).*

Model	EM	F1
BERT (original)	0.6160	0.7614
Multi-passage BERT	0.6082	0.7375
Multi-passage BERT + DL	**0.6296**	0.7728
Multi-passage BERT + DL + Aux	0.6287	**0.7742**

the self-adjusting dice loss help boosts the performance of the models, especially in question-context pairs having long length. Even though our model design is specifically aimed at longer passages, the performance of our method can also be observed in the group of questions with shorter context passages which is shown in the first row of Table 5. This suggests that our method is robust regardless of context passage length.

Table 5: *F1-score breakdown by the length of the context passage. The last column shows the improvement of our model over the original BERT.*

Group of passage (tokens)	BERT (Original)	Our Model	Improvement
0 – 1,000	0.7681	0.7781	0.0100
1,000 – 1,500	0.7521	0.7703	0.0181
2,000 +	0.7096	0.7407	0.0311

6. CONCLUSION

In this paper, we built a system designed specifically to handle question answering tasks in Thai having longer context passages. In our model, it is evident that the combination of (i) multi-passage BERT, (ii) self-adjusting DICE loss, and (iii) an auxiliary classification task leads to improvement in performance. The experiment was conducted on the Thai NSC QA dataset which contains more than 15,000 factoid questions. Our experiment demonstrates that our proposed model with all combinations performs better than the baseline BERT (from 0.6160, 0.7614 to 0.6287, 0.7742 in terms of exact match and F1, respectively). Our model can improve performance of F1 by 1% on a passage with less than 1,000 tokens. On longer passages with 2,000 tokens or more, it can improve F1 over by 3.11%.

References

[1] P. Rajpurkar, J. Zhang, K. Lopyrev and P. Liang, “SQuAD: 100,000+ Questions for Machine Comprehension of Text,” in *EMNLP*, 2016.

[2] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A.M. Dai, J. Uszkoreit, Q. Le and S.Petrov, “Natural Questions: A Benchmark for Question Answering Research,” *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453-466, 2019.

[3] M. Joshi, E. Choi, D. Weld and L. Zettlemoyer, “TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension,” *Association for Computational Linguistics Vancouver*, Canada, 2017.

[4] M. Dunn, L. Sagun, M. Higgins, V.U. Guney, V.Cirik and K. Cho, “SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine,” *Computation and Language*, arXiv:1704.05179, 2017.

[5] W. He, K. Liu, J. Liu, Y. Lyu, S. Zhao, X. Xiao,

Y. Liu, Y. Wang, H. Wu, Q. She, X. Liu, T. Wu and H. Wang, "DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications," *Association for Computational Linguistics*, Melbourne, Australia, 2018.

[6] K. Trakultaweekoon, S. Thaiprayoon, P. Palingoon and A. Rugchatjaroen, "The First Wikipedia Questions and Factoid Answers Corpus in the Thai Language," *2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, pp. 1-4, 2019.

[7] M. Seo, A. Kembhavi, A. Farhadi and H. Hajishirzi, "Bidirectional Attention Flow for Machine Comprehension," *Computation and Language*, arXiv:1611.01603, 2016.

[8] H.Y. Huang, C. Zhu, Y. Shen, W. and Chen, "FusionNet: Fusing via Fully-Aware Attention with Application to Machine Comprehension," *Computation and Language*, arXiv:1711.07341, 2017.

[9] J. Devlin, M.W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Association for Computational Linguistics*, Minneapolis, Minnesota, pp. 4171-4186, 2019.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser and I. Polosukhin, "Attention is All you Need," in *NIPS*, 2017.

[11] Z. Wang, P. Ng, X. Ma, R. Nallapati and B. Xiang, "Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) Association for Computational Linguistics*, Hong Kong, China, pp. 5878-5882, 2019.

[12] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu and J. Li, "Dice Loss for Data-imbalanced NLP Tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics Association for Computational Linguistics*, Online, pp. 465-476, 2020.

[13] S. Back, S. Yu, S.R. Indurthi, J. Kim and J. Choo, "MemoReader: Large-Scale Reading Comprehension through Neural Memory Controller," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing Association for Computational Linguistics*, Brussels, Belgium, pp. 2131-2140, 2018.

[14] Y. Wang,, K. Liu, J. Liu, W. He, Y. Lyu, H. Wu, S. Li and H. Wang, "Multi-Passage Machine Reading Comprehension with Cross-Passage Answer Verification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) Association for Computational Linguistics*, Melbourne, Australia, pp. 1918-1927, 2018.

[15] S. Wang, M. Yu, X. Guo, Z. Wang, T. Klinger, W. Zhang, S. Chang, G. Tesauro, B. Zhou and J. Jiang, "R3: Reinforced Reader-Ranker for Open-Domain Question Answering," *Computation and Language*, arXiv:1709.00023, 2017.

[16] L. Pang, Y. Lan, J. Guo, J. Xu, L. Su and X. Cheng, "HAS-QA: Hierarchical Answer Spans Model for Open-domain Question Answering," in *AAAI*, 2019.

[17] W. Jitkrittum, C. Haruechaiyasak and T. Theeramunkong, "QAST: Question Answering System for Thai Wikipedia (08/06)," 2009.

[18] A. Kongthon, S.Kongyoung, C. Haruechaiyasak and P. Palingoon, "A Semantic Based Question Answering System for Thailand Tourism Information," in *Proceedings of the KRAQ11 workshop Asian Federation of Natural Language Processing*, Chiang Mai, pp. 38-42, 2011.

[19] T. Noraset, L. Lowphansirikul and S. Tuarob, "WabiQA: A Wikipedia-Based Thai Question-Answering System," *Inf. Process. Manag.*, vol.58, pp. 102431, 2021.

[20] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[21] J. Ba, J.R. Kiros and G.E. Hinton, "Layer Normalization," *Machine Learning*, arXiv:1607.06450, 2016.

[22] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, 2016.

[23] Doshi, K., 2020. Transformers Explained Visually (Part 1): Overview of Functionality. In `https://towardsdatascience.com/transformers-explained-visually-part-1-\\overview-of-functionality-95a6dd460452`

[24] C. Clark and M. Gardner, "Simple and Effective Multi-Paragraph Reading Comprehension," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) Association for Computational Linguistics*, Melbourne, Australia, pp. 845-855, 2018.

[25] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel and N. Dehak, "Hierarchical Transformers for Long Document Classification," *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 838-844, 2019.

[26] W. Phatthiyaphaibun, K. Chaovavanich, C. Polpanumas, A. Suriyawongkul, L. Lowphansirikul and P. Chormai, "PyThaiNLP: Thai Natural Language Processing in Python," 2016.

[27] D.P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Machine Learning*, arXiv:1412.6980, 2015.

Theerit Lapchaicharoenkit received his Master degree from Department of Computer Engineering, Chulalongkorn University, Bangkok, Thailand. in 2020. During his graduate year, he conducted the research in the domain of artificial intelligence, natural language processing to be more specific. The focus of the research is in the area of natural language processing in Thai language.

Peerapon Vateekul received his Ph.D. degree from Department of Electrical and Computer Engineering, University of Miami (UM), Coral Gables, FL, U.S.A. in 2012. Currently, he is an associate professor at Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Thailand. His research falls in the domain of machine learning, data mining, deep learning, text mining, and big data analytics. To be more specific, his works include variants of classification (hierarchical multi-label classification), natural language processing, data quality management, and applied deep learning techniques in various domains, such as, healthcare, geoinformatics, hydrometeorology, etc. Some examples of AI-assisted medical diagnoses are real-time polyp detection from colonoscopy, gastrointestinal metaplasia segmentation from gastroscopy, dyssynergic defecation classification, depressive scoring from interview videos, Parkinson's face classification, movement disorder diagnosis, etc.