MRC Examples Answerable by BERT without a Question are Less Effective in MRC Model Training

Hongyu Li[†], **Tengyang Chen**[†], **Shuting Bai**[†], **Takehito Utsuro**[†], **Yasuhide Kawada**[‡]
[†]Graduate School of Systems and Information Engineering, University of Tsukuba, Japan
[‡]Logworks Co., Ltd., Japan

Abstract

Models developed for Machine Reading Comprehension (MRC) are asked to predict an answer from a question and its related context. However, there exist cases that can be correctly answered by an MRC model using BERT, where only the context is provided without including the question. In this paper, these types of examples are referred to as "easy to answer", while others are as "hard to answer", i.e., unanswerable by an MRC model using BERT without being provided the question. Based on classifying examples as answerable or unanswerable by BERT without the given question, we propose a method based on BERT that splits the training examples from the MRC dataset SQuAD1.1 into those that are "easy to answer" or "hard to answer". Experimental evaluation from a comparison of two models, one trained only with "easy to answer" examples and the other with "hard to answer" examples demonstrates that the latter outperforms the former.

1 Introduction

The Machine Reading Comprehension (MRC) task locates the best corresponding natural language answer when provided a question and its related context. In recent years, MRC models using neural networks have been proposed for SQuAD (Pranav et al., 2016, 2018), which is a large-scale, high-quality English MRC dataset. Most recent neural network based MRC models have outperformed human performance (Devlin et al., 2019).

Among those existing work, to analyze the difficulty of several popular MRC benchmarks such as bAbI (Weston et al., 2016), SQuAD (Pranav et al., 2016), CBT (Hill et al., 2016), CNN (Hermann et al., 2015) and Who-did-What (Onishi et al., 2016), Kaushik and Lipton (2018) established sensible baselines for these datasets, and found that question-only and context-only (which is called *passage-only* in Kaushik and Lipton (2018)) models often performs surprisingly well. In particular, context-only models achieve over 50% accuracy on 14 out of 20 bAbI tasks, and as for CBT, only the last one of the 20 sentences provided as a context is necessary to achieve a comparable accuracy. They also indicated that SQuAD is designed more carefully than other datasets and achieved F1 scores of only 4% and 14.8% respectively on question-only and context-only models, which are relatively lower. Kaushik and Lipton (2018) demonstrated that published MRC datasets should characterize the level of difficulty, and specifically, the extent to which questions and contexts are essential. Moreover, Kaushik and Lipton (2018) also claimed that follow-up papers reporting improvements ought to report performance both on the full task and variations omitting questions and contexts. In view of the point demonstrated in Kaushik and Lipton (2018), we concentrate more on the difficulty of every single MRC example, and aim to split the examples into easy ones and hard ones.

Given the MRC dataset SQuAD1.1 (where each MRC example denoted as the tuple $\langle Q, C, A \rangle$ of the question Q, the context C, and the answer A) and the fine-tuned MRC model using BERT (Devlin et al., 2019), there exist *context-only* examples that can be correctly answered, where only the context is provided without including the question. By focusing on this fact, this paper proposes a method that splits the MRC examples into binary classes of "easy to answer" or "hard to answer". A 10-fold cross-validation was applied on approximately 87,600 SQuAD1.1 training examples comprised of 12,500 "easy to answer" and 75,000 "hard to answer" classes. From the comparison of the two classes, the followings are two significant findings. (1) Based

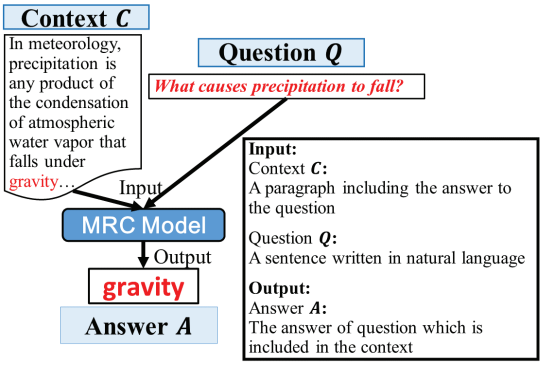

Figure 1: An MRC Model using Neural Networks

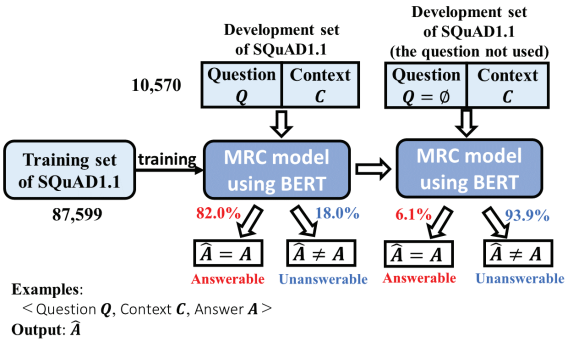

Figure 2: Detecting MRC Examples as Answerable without an Input Question

on the performance of training the BERT MRC models with 12,500 "easy to answer" and "hard to answer" examples each, the model trained with "hard to answer" examples outperformed the other. (2) An analysis of the position distribution of answers A within the context C, answers from the "easy to answer" MRC example class tend to be located around the beginning of the context compared with those from the "hard to answer" MRC example class.

2 Machine Reading Comprehension using Neural Networks

Figure 1 shows the framework of MRC models that use neural networks. In the MRC model, when a question and context are input, the starting and ending positions of the answer with respect to the question within the context are predicted.

Let ts be a set of test examples with each example denoted as $s(\in ts)$. Here, s is represented as $s = \langle Q, C, A\rangle$. Also, if a set of examples for training MRC models is denoted as tr, then the corresponding model is represented as $m(tr)$. Then, the answer $\widehat{A}$ predicted from an input test example s with the trained MRC model $m(tr)$ is denoted as

$$\widehat{A} \;=\; answer\Big(m(tr),\; s\Big)$$

A Boolean predicate $answerable$ classifies if the given test example s is "answerable" or "unanswerable" by the trained MRC model $m(tr)$, and is defined according to if the predicted answer $\widehat{A}$ is the same as the reference answer A as

$$answerable\Big(m(tr),\; s\Big) \;=\; \begin{cases} 1 & (\widehat{A} = A) \\ 0 & (\widehat{A} \neq A) \end{cases}$$

3 Machine Reading Comprehension using BERT

As described above, the MRC models are trained and tested with 87,600 training examples from the MRC dataset SQuAD1.1. The fine-tuning module for machine reading comprehension[1] was applied to the pre-trained Multilingual Cased model[2] of BERT.[3]

4 MRC Examples Answerable without a Question

First, we evaluate the BERT MRC model trained with the 87,600 training examples against 10,570 development examples from of SQuAD1.1. As shown in Figure 2, we compare the two cases of *with* or *without* the question for each of these development examples. Let s denote one of the 10,570 development examples represented as $s = \langle Q, C, A\rangle$, as before, which are each evaluated with the BERT MRC model trained with the 87,600 examples. From these results, 82% are correctly answered and classified as "answerable" with the remaining 18% incorrectly answered and classified as "unanswerable." Next, let s' denote an MRC example obtained by replacing the question Q from each example s of the development examples with an empty question $Q' = \emptyset$ represented as $s' = \langle Q' = \emptyset, C, A\rangle$. After evaluating the trained BERT MRC model with each example s', as shown on the right of Figure 2, 6.1% are

[1] `run_squad.py`, with the number of epochs as 2, batch size as 8, learning rate as 0.00003, and the maximum sequence length as 320.

[2] Trained with 104 languages, available from https://github.com/google-research/bert/blob/master/multilingual.md.

[3] The TensorFlow version of BERT (https://github.com/google-research/bert) is used.

		Training set	
		Questions used tr_i	Questions not used $tr_i{}^{Q=\emptyset}$
Test set	questions used $s\ (\in ts_i)$	Answerable examples $a^1(ts_i)$ Unanswerable examples $ua^1(ts_i)$ $\lvert a^1\rvert = 60,668$ $\lvert ua^1\rvert = 26,931$	Answerable examples $a^2(ts_i)$ Unanswerable examples $ua^2(ts_i)$ $\lvert a^2\rvert = 11,498$ $\lvert ua^2\rvert = 76,101$
	questions not used $s^{Q=\emptyset}$ $(s \in ts_i)$	Answerable examples $a^3(ts_i)$ Unanswerable examples $ua^3(ts_i)$ $\lvert a^3\rvert = 3,682$ $\lvert ua^3\rvert = 83,917$	Answerable examples $a^4(ts_i)$ Unanswerable examples $ua^4(ts_i)$ $\lvert a^4\rvert = 10,008$ $\lvert ua^4\rvert = 77,591$

Table 1: Splitting the MRC Examples into "Answerable" and "Unanswerable" Examples with the Corresponding Statistics

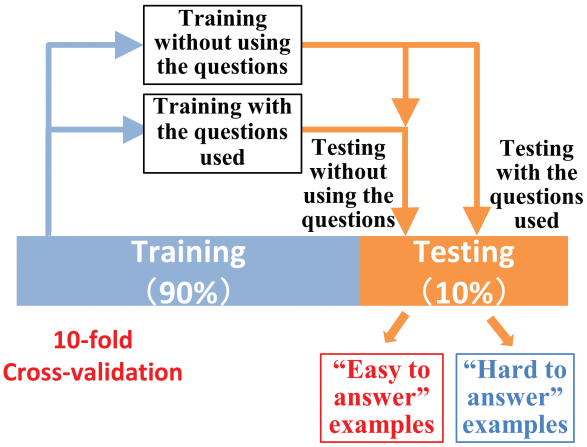

Figure 3: Splitting the MRC Examples into "Easy to Answer" and "Hard to Answer" Classes with 10-fold Cross-Validation

correctly answered and classified as "answerable" even without being provided an appropriate question, while the remaining 93.9% are incorrectly answered and classified as "unanswerable."

5 Splitting MRC Examples into "Easy to Answer" and "Hard to Answer" Classes

Following the procedure for detecting MRC examples as answerable without a question demonstraed in the previous section, we similarly split the 87,600 SQuAD1.1 training examples into "easy to answer" and "hard to answer" classes. As illustrated in Figure 3, the process designates 10% of the examples as "easy to answer" and "hard to answer" classes for testing through one fold of 10-fold cross-validation, which is repeated ten times, resulting in 12,500 "easy to answer" and 75,100 "hard to answer" classes. From this, we obtain the following three types of evaluation results.

(i) The MRC model is trained with the training examples that include questions used as they are, while the trained MRC model is evaluated against the MRC test examples without questions.

(ii) The MRC model is trained with training examples that do not include questions, with

(ii-a) the trained MRC model is evaluated against the MRC test examples without questions, or

(ii-b) the trained MRC model is evaluated against the MRC test examples with questions.

Detailed Procedure

The SQuAD1.1 dataset is composed of approximately 100,000 MRC examples that use 23,215 paragraphs extracted from 536 Wikipedia articles as context. With these contexts, questions and answers are annotated through crowdsourcing to generate the complete 100,000 MRC example set. From these examples, we apply N-fold cross-validation (N=10 in this paper) to the set U of the MRC training examples collected from 442 out of the 536 Wikipedia articles.

Before the N-fold cross-validation, we first divide the 442 Wikipedia articles into disjoint N subsets. From the i-th ($i = 1, \ldots, N$) subset of Wikipedia articles, we obtain the i-th test set ts_i of the MRC examples, and the i-th training set of the MRC examples is obtained as the set tr_i of the remaining MRC examples. Then, the set U of the complete SQuAD1.1 training example set is represented as

$$U = \bigcup_{i=1,\ldots,N} ts_i \quad \left(ts_i \cap ts_j = \emptyset \ \ (i \neq j) \right)$$

As shown in Table 1, from the i-th training set tr_i of the MRC examples, each of which contains a question, another training set $tr_i{}^{Q=\emptyset}$ of the MRC examples is obtained by removing the question Q from each example. So, each MRC example in the obtained training set $tr_i{}^{Q=\emptyset}$ now has an empty question. Similarly, from a test MRC example s in the i-th test set ts_i of the MRC examples that contains a question, another test MRC example $s^{Q=\emptyset}$ is obtained by removing its question Q from s. So, the obtained test MRC example $s^{Q=\emptyset}$ has an empty question. By pairing the two training sets tr_i and $tr_i{}^{Q=\emptyset}$ from the MRC examples and the two test MRC examples s and $s^{Q=\emptyset}$, as shown

in Table 1, a resulting four pairs of training sets from the MRC examples and a test MRC example can be examined as to if the MRC model trained with the designated training set is "answerable" or "unanswerable" given the designated test MRC example. Finally, in each of these four pairs, the set ts_i of the test MRC examples is split into the set $a^{\alpha}(ts_i)$ of answerable test MRC examples and $ua^{\alpha}(ts_i)$ of unanswerable test MRC examples, according to $(\alpha = 1, 2, 3, 4)$

$$a^1(ts_i) = \left\{ s \in ts_i \middle| answerable\Big(m(tr_i), s\Big) = 1 \right\}$$

$$a^2(ts_i) = \left\{ s \in ts_i \middle| answerable\Big(m({tr_i}^{Q=\emptyset}), s\Big) = 1 \right\}$$

$$a^3(ts_i) = \left\{ s \in ts_i \middle| answerable\Big(m(tr_i), s^{Q=\emptyset}\Big) = 1 \right\}$$

$$a^4(ts_i) = \left\{ s \in ts_i \middle| answerable\Big(m({tr_i}^{Q=\emptyset}), s^{Q=\emptyset}\Big) = 1 \right\}$$

$$ua^{\alpha}(ts_i) = ts_i - a^{\alpha}(ts_i) \ (\alpha = 1, 2, 3, 4)$$

The sets $a^{\alpha}(ts_i)$ $(\alpha = 1, 2, 3, 4)$ of "answerable" test MRC examples are obtained by evaluating the MRC model trained with the training sets tr_i (with questions) or ${tr_i}^{Q=\emptyset}$ (without questions) against s (with a question) or $s^{Q=\emptyset}$ (without a question). We define the set E of "easy to answer" MRC examples as the union of the three sets $a^{\alpha}(ts_i)$ $(\alpha = 2, 3, 4)$ of "answerable" test MRC examples. For these, we collect the "answerable" test MRC examples over the cases with questions removed either from the training or test MRC examples ($a^1(ts_i)$ is excluded because the questions are used in both the training and test MRC examples). The set H of "hard to answer" MRC examples is subsequently defined as the complement set of E.[4]

Consequently, as shown in Table 2, the set U of the complete SQuAD1.1 training examples is split into the set E of 12,487 "easy to answer" examples and the set H of 75,112 "hard to answer" examples. Figure 4 compares the distributions of positions of answers within the contexts as a ratio of the "start position of an answer" to the "length of context". These results indicate that the answers of the "easy to answer" MRC examples tend to be located near the beginning of the context as compared with those of the "hard to answer" MRC examples.[5] We repeat this splitting procedure ten times and compare the numbers of "easy to answer" and "hard to answer" examples, where we have almost the same results as we report in this section. For examples of the "easy to answer" MRC examples, Figure 5 provides two cases, one of which is a typical "easy to answer" with its answer located exactly at the beginning of the context, and a second as the opposite class with its answer located exactly at the end of the context.

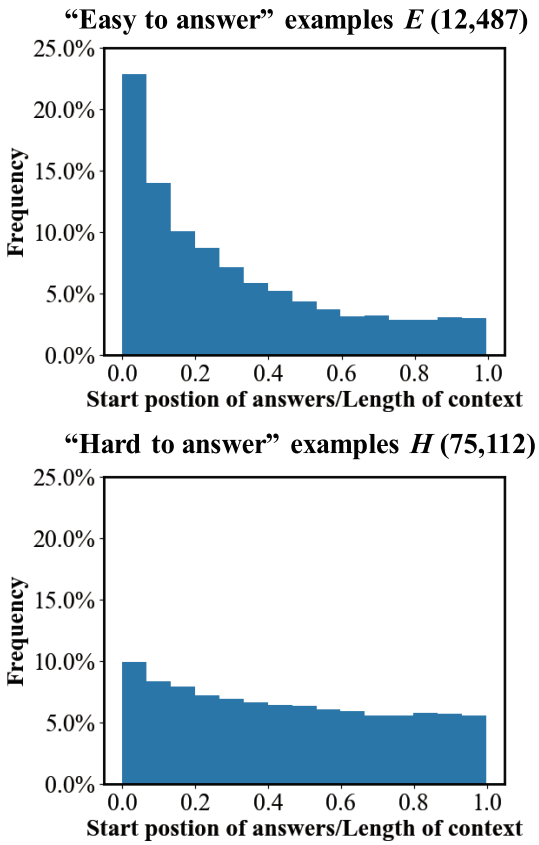

Figure 4: Distributions of the Positions of Answers in the Contexts as the Ratio of "start position of an answer" / "length of context"

6 Effectiveness of "Hard to Answer" Examples in MRC Model Training

We next evaluate the effectiveness of "hard to answer" and "easy to answer" MRC examples based

[4] Over the set U of the complete SQuAD1.1 training examples, the set a^{α} of "answerable" examples and the set ua^{α} of "unanswerable" examples are defined as $a^{\alpha} = \bigcup_{i=1,\ldots,N} a^{\alpha}(ts_i)$, $ua^{\alpha} = U - a^{\alpha}$ $(\alpha = 1, 2, 3, 4)$, where the number of examples in each set is provided in Table 1.

[5] We also compare the context length between the "easy to answer" and "hard to answer" examples, where we did not detect any significant difference.

Question Q : What began as an almost exclusively linguistic and philological enterprise?
Context C : ***Slavic studies*** began as an almost exclusively linguistic and philological enterprise. As early as 1833, Slavic languages were recognized as Indo-European.
Answer A : ***Slavic studies***

Question Q : What is a significant early architectural canonical type?
Context C : Texts on architecture have been written since ancient time. These texts provided both general advice and specific formal prescriptions or canons. Some examples of canons are found in the writings of the 1st-century BCE Roman Architect Vitruvius. Some of the most important early examples of canonic architecture are ***religious***.
Answer A : ***religious***

Figure 5: Two Sample "Easy to Answer" MRC Examples

Training set	Number of examples
U: training set of SQuAD1.1	87,599
H: "hard to answer" examples	75,112
M: examples randomly sampled from U	
E: "easy to answer" examples	12,487
H_{sml}: examples randomly sampled from H	
M_{sml}: examples randomly sampled from U	

Table 2: Number of Examples in Each Training Set

on the performance of each class when used for the MRC model training. The sets shown in Table 2 are evaluated as the MRC model training examples. In addition to the sets H and E, we evaluate a set M comprised of $|H| = 75,112$ MRC examples randomly sampled from U and sets H_{sml} and M_{sml} of $|E| = 12,487$ MRC examples randomly sampled from H and U, respectively. The sets H_{sml} and M_{sml} are intended to directly compare the effectiveness of the "easy to answer", "hard to answer", and (randomly sampled) SQuAD1.1 training examples by restricting the numbers of the training examples to be the same. The set M is also intended to directly compare the effectiveness of the "hard to answer" and SQuAD1.1 training examples by restricting the numbers of training examples to be the same. All these sets are used to fine-tune the BERT pre-trained model on the MRC task, and the development set of SQuAD1.1

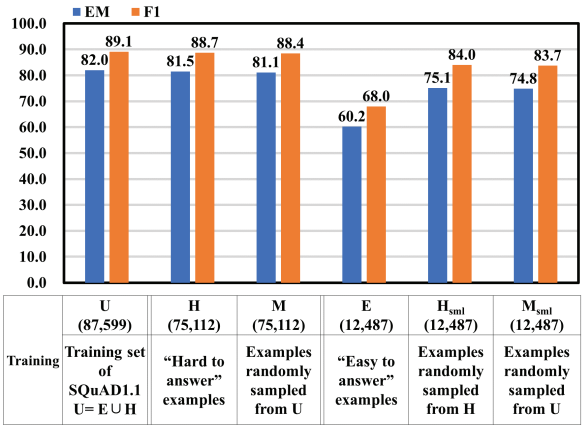

Figure 6: Evaluation Results on the Development Set of SQuAD1.1 where EM is an Exact Match, and F1 is the Macro-Average of the F1 Score per Example

is used as the test set for each evaluation. For the evaluation measures, we utilize the exact match (EM), which is defined as the rate of examples with a predicted answer that exactly matches the reference answer. The macro average of the F1 score is calculated from the precision and recall between the token sequences of the predicted and reference answers.

Figure 6 compares the performance of the five MRC training examples, and Figure 7 presents the learning curves with the training examples of the sets E and H_{sml} of Table 2 used against the development set of SQuAD1.1 as the test set. From both results, the set H_{sml} outperforms the set E with a statistically significant (p<0.01) difference, suggesting that the "hard to answer" examples are effective in MRC model training[6]. Unfortunately, however, Figure 6 also presents that the performance of H_{sml} is almost comparable with that of M_{sml}. From this result, our definite future work includes inventing a technique of automatic selection of MRC training examples from the set U of the complete SQuAD1.1 training example set, which outperform those of the same size randomly sampled from U.

Also, although we omit the detailed evaluation results, in addition to BERT, we also applied SpanBERT (Joshi et al., 2020)[7] (base & cased) and XLNet (Yang et al., 2019)[8] (XLNet-Large, Cased) and obtained the similar results regarding both of

[6]We repeat the splitting procedure and the evaluation procedure ten times, where we have almost the same evaluation results we report in this section.

[7]https://github.com/facebookresearch/SpanBERT

[8]https://github.com/zihangdai/xlnet

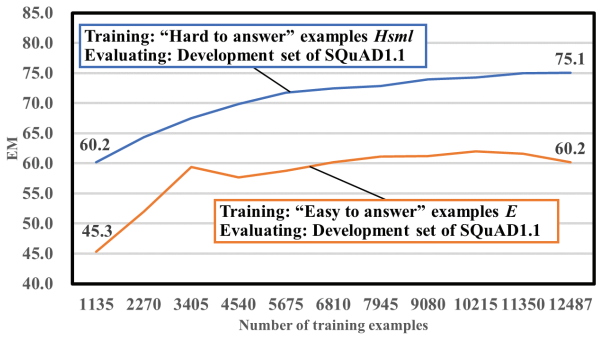

Figure 7: Comparison of the Learning Curves for the Exact Match (EM)

(1) the model trained with "hard to answer" examples outperformed that trained with "easy to answer" ones, and (2) answers from the "easy to answer" MRC example class tend to be located around the beginning of the context compared with those from the "hard to answer" MRC example class.

7 Related Work

Swayamdipta et al. (2020) proposed a general framework of identifying three regions, namely, ambiguous, easy to learn, and hard to learn within a dataset, and applied the framework to several tasks such as natural language inference and sentence-level machine reading comprehension. It is concluded that ambiguous instances are useful for high performance, easy to learn instances are aid optimization, and hard to learn instances correspond to data errors. Following the conclusions of Swayamdipta et al. (2020), our future work include applying the framework of Swayamdipta et al. (2020) to the tasks of machine reading comprehension studied in this paper and investigating the difference of our notion of "easy to answer" / "hard to answer" and their notion of "easy to learn" / "hard to learn." Among other related work, Sugawara et al. (2018) studied splitting 12 MRC datasets into easy and hard subsets according to two types of simple lexical based heuristics and showed that the performance against easy subsets were lower than the whole datasets. Min et al. (2018) also studied to select minimal set of sentences within the context of existing MRC datasets to answer the MRC question.

In the task of recognizing textual entailment that classifies the relation between a pair of two sentence as a premise and hypothesis, Tsuchiya (2018) compared two of the "Recognizing Textual Entailment" datasets, SICK (Bowman et al., 2015) and SNLI (Marelli et al., 2014). Tsuchiya reported that the cases of SNLI had the correct textual entailment labels predicted when only the hypothesis sentence was provided and without the premise sentence. However, Tsuchiya (2018) also pointed out that, a hidden bias in the SNLI corpus caused much of the high accuracy achieved by the neural network based models that were trained with SNLI.

Developing machine reading comprehension datasets requires an expensive and time-consuming effort to manually create questions from paragraphs and extract spans of text from each paragraph to represent the answer to each question. The approach of active learning, in which the key idea is that a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns (Settles, 1995, 2010), could be applied to reduce the cost of developing MRC datasets. While there exists no previous study that applies the active learning technique for machine reading comprehension task, other work applied the technique to reduce the cost of developing datasets for other NLP tasks (Sener and Savarese, 2018; Chen et al., 2019), image classification (Beluch et al., 2018; Fang et al., 2017), as well as other machine learning tasks, such as predicting molecular energetics in the field of chemistry (Smith et al., 2018).

8 Conclusion

We proposed a method based on BERT (Devlin et al., 2019) that splits the training examples from the MRC dataset SQuAD1.1 into classes of "easy to answer" and "hard to answer." Experimental evaluations of comparing the two models, one of which is trained only with the "easy to answer" examples and the other with the "hard to answer" examples, demonstrate that the latter outperformed the former. Future work includes applying the analysis procedure of this paper to several popular MRC benchmark datasets other than SQuAD (Pranav et al., 2016) and investigating whether the similar results are obtained. We also work on deeper analysis of the characteristics of "easy to answer" / "hard to answer" examples to find out features that are related to the disparity of training effectiveness.

References

W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler. 2018. The power of ensembles for active learning in image classification. In *Proc. 31st CVPR*, pages 9368–9377.

S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proc. 20th EMNLP*, pages 632–642.

X. C. Chen, A. Sagar, J. T. Kao, T. Y. Li, C. Klein, S. Pulman, A. Garg, and J. D. Williams. 2019. Active learning for domain classification in a commercial spoken personal assistant. In *Proc. 20th INTERSPEECH*, pages 1478–1482.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*, pages 4171–4186.

M. Fang, Y. Li, and T. Cohn. 2017. Learning how to active learn: A deep reinforcement learning approach. In *Proc. EMNLP*, pages 595–605.

K. M. Hermann, T. Kociskỳ, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. 2015. Teaching machines to read and comprehend. In *Proc. 28th NIPS*, pages 1693–1701.

F. Hill, A. Bordes, S. Chopra, and J. Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. In *Proc. 4th ICLR*, pages 1–13.

M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

D. Kaushik and Z. C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proc. EMNLP*, pages 5010–5015.

M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proc. 9th LREC*, pages 216–223.

S. Min, V. Zhong, R. Socher, and C. Xiong. 2018. Efficient and robust question answering from minimal context over documents. In *Proc. 56th ACL*, pages 1725–1735.

T. Onishi, H. Wang, M. Bansal, K. Gimpel, and D. McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *Proc. EMNLP*, pages 2230–2235.

R. Pranav, Z. Jian, L. Konstantin, and L. Percy. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proc. EMNLP*, pages 2383–2392.

R. Pranav, J. Robin, and L. Percy. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proc. 56th ACL*, pages 784–789.

O. Sener and S. Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *Proc. 6th ICLR*, pages 1–13.

B. Settles. 1995. Active learning literature survey. *Science*, 10(3):237–304.

B. Settles. 2010. Active learning literature survey. University of Wisconsin–Madison, Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev, and A. E. Roitberg. 2018. Less is more: Sampling chemical space with active learning. *Journal of Chemical Physics*, 148(241733).

S. Sugawara, K. Inui, S. Sekine, and A. Aizawa. 2018. What makes reading comprehension questions easier? In *Proc. EMNLP*, pages 4208–4219.

S. Swayamdipta, R. Schwartz, N. Lourie, Y. Wang, H. Hajishirzi, N. A. Smith, and Y. Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proc. EMNLP*.

M. Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proc. 11th LREC*, pages 1506–1511.

J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov. 2016. Towards AI-complete question answering: A set of prerequisite toy tasks. In *Proc. 4th ICLR*, pages 1–14.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Proc. 33rd NeurIPS*, pages 5753–5763.