Computers, Materials & Continua
DOI: 10.32604/cmc.2023.031983
Article

Information Extraction Based on Multi-turn Question Answering for Analyzing Korean Research Trends

Seongung Jo[1], Heung-Seon Oh[1,*], Sanghun Im[1], Gibaeg Kim[1] and Seonho Kim[2]

[1]School of Computer Science and Engineering, KOREATECH, Cheonan, 31253, Korea
[2]Korea Institute of Science and Technology Information, KISTI, Daejeon, 34141, Korea
*Corresponding Author: Heung-Seon Oh. Email: ohhs@koreatech.ac.kr
Received: 02 May 2022; Accepted: 01 August 2022

Abstract: Analyzing Research and Development (R&D) trends is important because it can influence future decisions regarding R&D direction. In typical trend analysis, topic or technology taxonomies are employed to compute the popularities of the topics or codes over time. Although it is simple and effective, the taxonomies are difficult to manage because new technologies are introduced rapidly. Therefore, recent studies exploit deep learning to extract pre-defined targets such as problems and solutions. Based on the recent advances in question answering (QA) using deep learning, we adopt a multi-turn QA model to extract problems and solutions from Korean R&D reports. With the previous research, we use the reports directly and analyze the difficulties in handling them using QA style on Information Extraction (IE) for sentence-level benchmark dataset. After investigating the characteristics of Korean R&D, we propose a model to deal with multiple and repeated appearances of targets in the reports. Accordingly, we propose a model that includes an algorithm with two novel modules and a prompt. A newly proposed methodology focuses on reformulating a question without a static template or pre-defined knowledge. We show the effectiveness of the proposed model using a Korean R&D report dataset that we constructed and presented an in-depth analysis of the benefits of the multi-turn QA model.

Keywords: Natural language processing; information extraction; question answering; multi-turn; Korean research trends

1 Introduction

Recently, technologies have advanced rapidly and simultaneously due to in-domain achievements and domain convergences. Identifying technology trends [1,2] is essential for business and research aspects. Technology trends can be identified by analyzing R&D trends because the nature of research is that it improves technologies. Utilizing topic or technology taxonomies such as International Patent

Classification (IPC)[1] and Cooperative Patent Classification (CPC)[2] in patents, where several codes in taxonomy are assigned to documents, is a popular way to analyze research trends [3–5]. However, it is difficult to construct and manage a target taxonomy using these methods because the structure becomes large and complex as new codes are introduced continuously. Moreover, trends are very abstract even with a fine-grained taxonomy because occurrence counts of a code represent them over time in general.

One way to solve the aforementioned problems is to exploit IE methods [6–10] for pre-defined targets such as problem and solution. Recent IE methods [11–13] actively employ machine reading comprehension (MRC), and QA techniques because they achieved significant improvements using neural language models (NLMs) such as BERT [14] and are easily adaptable to other Natural Language Processing (NLP) tasks with minor modifications, i.e., reformulating a source example to a triple of a question, context, and answer.

In all R&D projects funded by the South Korean government, a final report describing problems and corresponding solutions must be submitted as part of the deliverables. We can effectively support future decisions regarding an R&D direction if the trends of both problems and solutions are identified automatically using MRC QA methods. We observed two characteristics in Korean R&D reports as shown in Fig. 1. First, several problems and solutions can appear in a report because an R&D project consists of several components to achieve a goal over the years. Second, among problems and solutions, there exist M:1 and 1:N relations as well as 1:1 because a problem can be dealt with using various solutions and vice versa.

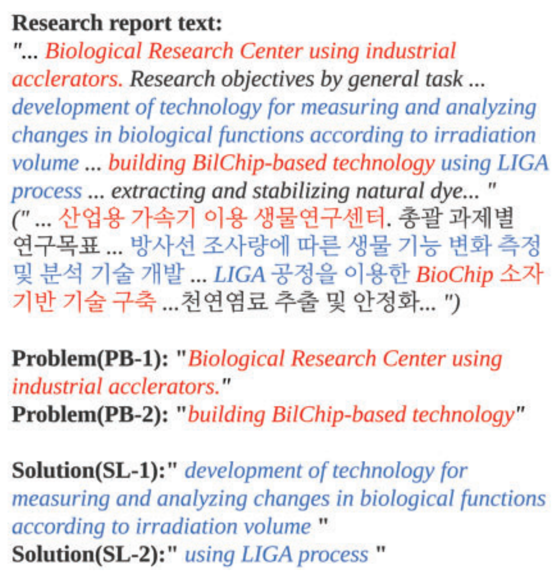

Figure 1: Example of a Korean R&D report

Because of the aforementioned characteristics, recent single-turn MRC QA models outputting an answer for a pair of a question and context as an input at once using pre-trained NLMs [14–17] have difficulties in extracting problems and solutions. First, the numbers of problems and solutions are unknown in advance. Second, the same problems and solutions can be extracted repeatedly from a

[1]https://www.wipo.int/classifications/ipc/en/
[2]https://www.cooperativepatentclassification.org/index

report, although the exact numbers are given. Third, M:1 and 1:N relations cannot be resolved because the nature of single-turn QA models only considers 1:1 relation. This paper focuses on the first and second problems, whereas the third is a type of relation extraction problem, which we reserve for our future work.

Recently, several studies adopted multi-turn QA, a generalization of single-turn QA to a sequence of interactive input and output, in entity-relation and event extraction tasks [18–21] on ACE04 and CONLL04 datasets. They impose dependencies in a sequence of turns, where a downstream turn is constructed to extract a target (entity, relation, or event) using the results of a current turn. However, they cannot deal with multiple extractions appropriately where the types and numbers of targets are not fixed and repeated extractions where the same target can be extracted repeatedly in a document because each input turn is constructed for a different purpose with a pre-defined template. Thus, a downstream turn stops or goes wrong within pre-defined turns if a mistake occurs in a current turn. In addition, they do not consider repeated extractions because the tasks allow the repetition of a target and focus on a sentence-level extraction scope.

We propose a multi-turn QA model to extract problems and solutions using Korean R&D reports. Our model, equipped with question reformulation (QR) and downstream turn detection (DTD), can deal with multiple and repeated extractions of targets properly at a document level. In QR, a question is reconstructed using a history of the previous turns. Therefore, the model avoids extracting previously extracted same targets. DTD determines whether to perform further extractions or not without the given number of targets. Our model uses a discriminator, a BERT-like NLM [17] trained using a Generative Adversarial Network (GAN) architecture. The discriminator is trained through three phases with different purposes. First, it is pre-trained to generate general representations with a large volume of text data. Second, it is fine-tuned to accommodate QA representations on a benchmark Korean QA dataset (i.e., KorQuAD [22]). Third, the full QA model equipped with the discriminator is fine-tuned to extract multiple problems and solutions on a Korean R&D report dataset we built. The contributions of this paper are summarized as follows:

- We propose a multi-turn QA model to extract problems and solutions from the Korean R&D report dataset. Our model equipped with DTD and QR can deal with multiple and repeated extractions of problems and solutions appropriately.
- We show the effectiveness of the proposed model on the Korean R&D report dataset we constructed and present an in-depth analysis of the benefits of the multi-turn QA model.

The rest of this paper is organized as follows. Section 2 discusses related works to recent QA models.

In Section 3, we introduce our multi-turn QA model in detail. Section 4 presents the experimental results and in-depth error analysis of the Korean R&D report dataset we built. In Section 5, we conclude with a discussion and future research prospects.

2 Related Works

Several studies focused on extracting problems and solutions as targets in documents [6–10]. The critical point of extracting targets lies in precisely understanding the context of surrounding targets. A line of research [23–27] focuses on a general single or multi-passage MRC task extracting targets using QA with triplets consisting of a question, context, and answer. Therefore, most MRC models can be simplified to text span extraction tasks for extracting answers as targets with a given passage and question. Recently, NLMs [14–17] using Transformer [28] have shown an outstanding ability to contextualize by performing well in the MRC tasks.

With the outperformed results produced by NLMs in the MRC tasks, research on text extraction that tends to use a QA model for non-QA NLP tasks surfaced. References [12,13] are the first studies to get significant results using a single-turn QA model for multi-task learning, and coreference resolution. In addition, [18–21] focused on the methods that allow only one question to be asked to a target and cast as a multi-turn QA task for multiple target extractions. They showed meaningful entity-relation and event extraction results by composing multi-turn with more complex extraction scenarios and dependent question templates.

Our work, highly inspired by [18], focuses on extracting multiple problems and solutions as multi-turn QA tasks. In this paper, we extract the multiple targets properly and avoid the repeated targets simultaneously on the Korean R&D reports. We show that our multi-turn QA model can solve the aforementioned restrictions.

3 Proposed Extraction Model

Fig. 2 shows the overall training procedure of our multi-turn QA model. It uses the discriminator [17] trained using a GAN architecture as the encoder. The discriminator is trained through three phases with different purposes. First, it is pre-trained to construct general representations with a large amount of text data. Pretraining NLM is a critical task because it significantly affects the performance of downstream tasks. Because of this, we opted for KoELECTRA[3], a public version of ELECTRA [17] trained on a large amount of Korean text. Second, the discriminator with span prediction is fine-tuned to accommodate QA representations on a benchmark Korean QA dataset [22]. The span prediction, regarded as a decoder, is learned to predict the start and end positions of the target. This phase is essential before moving to the next phase because directly training the model on the dataset we built is insufficient because of the small size. This phase coincides with training a general single-turn QA model. Third, a full QA model is fine-tuned to extract problems and solutions on the Korean R&D report dataset we built. Compared to the previous phase, the decoder is composed of span prediction and DTD. The details of single and multi-turn QA models will be described.

3.1 Single-turn QA Model

Let's define the data format for the single-turn QA model in general. The dataset is composed of N examples $\{E_1, \ldots, E_N\}$ where each example E_i is a triple question q_i, context c_i, and answer a_i. Then, a QA model is trained to extract $\hat{a}_i = a_i$ with a given pair (c_i, q_i). The second phase of fine-tuning, shown in Fig. 2, coincides with this on the Korean QA dataset. The QA model consists of a discriminator and a span prediction module. The span prediction predicts the positions of the start and end tokens of the span in a context.

In our task on the R&D report dataset, c_i is a concatenation of the title and abstract in the report. q_i denotes a question prompt describing the type (one of problem or solution) to be extracted from the report, i.e., "What is [type] in context?". In general, the question prompt changes because there are various information needs against a context. However, it is fixed in our task as they are constrained to problems and solutions. a_i is one of the problems or solutions appearing in c_i, depending on q_i. Because of this setting, a single-turn QA model cannot deal with the repeated and multiple extractions

[3]https://github.com/monologg/KoELECTRA

of problems and solutions appropriately, even using state-of-the-arts (SOTA) NLMs such as BERT [14] and GPT-3 [29].

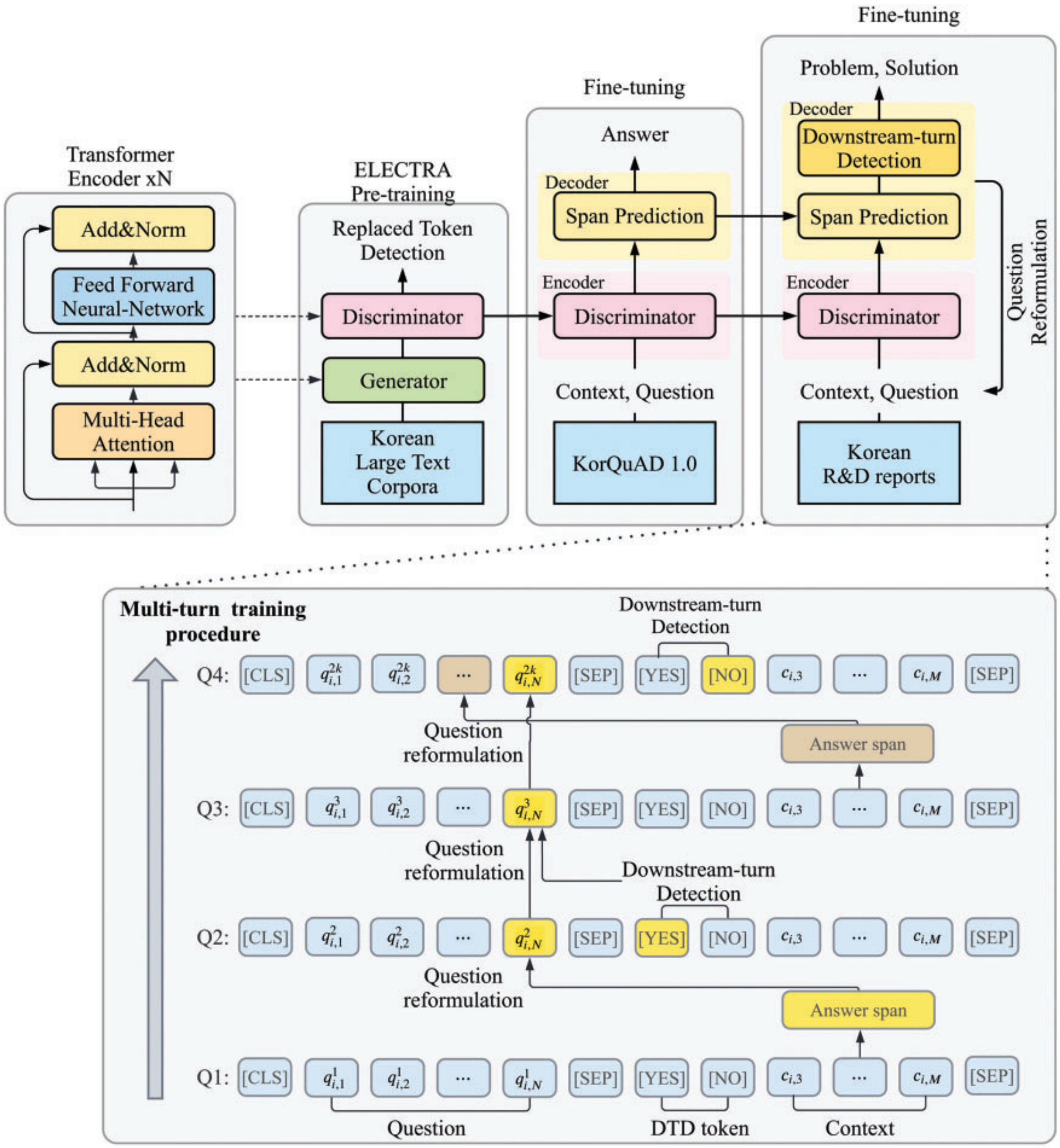

Figure 2: Overall training procedure of multi-turn QA model

3.2 Multi-turn QA Model

Our model is inspired by research on entity-relation and event extractions adopting multi-turn QA [18,19]. As shown in Fig. 2, DTD and QR over the discriminator are key components of the multi-turn QA. We start by defining a dataset consisting of N examples $\{E_1, \ldots, E_N\}$ for a multi-turn QA. Each example E_i is composed of a set of sub-examples $\left\{E_i^1, \ldots, E_i^{2K}\right\}$ where each E_i^k contains a pair of question q_i^k and answer a_i^k in the same context c_i and K is the number of targets to be extracted in a context c_i. A sub-example is used for extracting a target if k is odd, whereas it is used for determining a downstream turn if k is even. For DTD, two special tokens, "[YES]" and "[NO]", were inserted at the beginning of the context c_i. Fig. 3 shows the procedures of extracting problems (left) and solutions (right) with questions and answers in DTD and QR.

$$\mathbf{QR}\,(H,t,k) = \begin{cases} q_{DTD}\,(H,t) & \text{if } i = -1 \\ q_{SP}\,(H,t,k) & \text{otherwise} \end{cases} \tag{1}$$

$$\mathbf{DTD}\left(\hat{a}, th_i, \delta\right) = \begin{cases} \text{True} & \text{if } \hat{a} = \text{"}YES\text{"} \text{ or } \left(\delta < th_i \text{ and } \hat{a} = \text{"}NO\text{"}\right) \\ \text{False} & \text{otherwise} \end{cases} \tag{2}$$

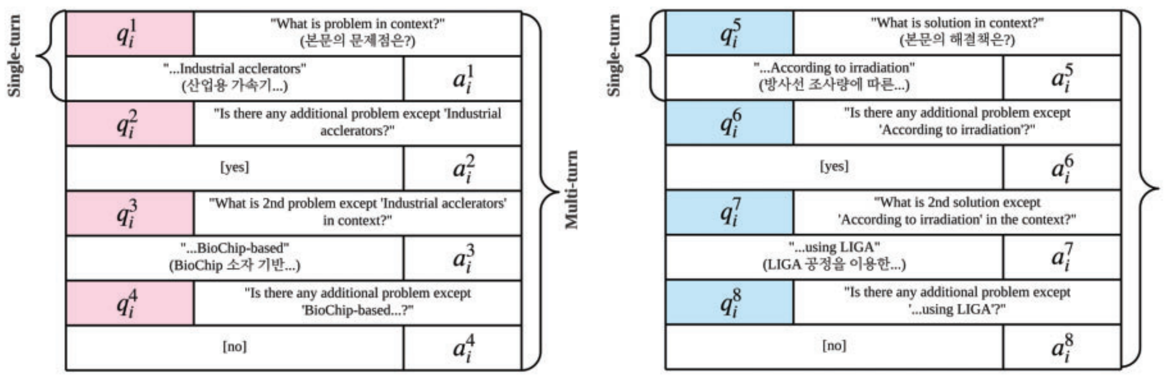

Figure 3: Procedures of extracting problems (left) and solutions (right) using a multi-turn QA model from a Korean R&D report shown in Fig. 1

In QR, a question prompt using Eq. (1) is constructed for extracting the target and determining a downstream turn by incorporating a history of the previous extractions H as shown in Tab. 1. i.e., "Is there any additional t except [H]?" and "What is [k] th [t] except [H] in the context?", respectively. Particularly, it can prevent the repeated extractions of the same problems and solutions because the question prompt provides clear information not to extract to a model. DTD using Eq. (2) determines whether to perform further extraction using the output of a QA model with a question prompt constructed using QR. It continues to perform the extraction if the output is "Yes". In the case of "No", we employ a threshold to filter false negative examples, denoted as threshold-based answer verification (TAV). The threshold values were determined using a validation dataset. Algorithm 1 describes the details of the inference procedure.

Table 1: Question prompt type by QR

Type	Text format
q_{DTD}	"Is there any additional [t] except [H]?"
q_{SP}	"What is [k th] [t] [except H] in the context?"

Algorithm 1: Inference procedure of our model

```
Input  : context and target type of the one report: c, t
Output: problems or solutions list: P

t: PB or SL
δ: confidence score
K: max turn

H ← []
k ← 1
while k ≠ K do
    q̇ ← QR(H, t, k) // SP
    â ← model(c, q̇)
    â append to H
    q̈ ← QR(H, t, −1)// DTD
    â, δ ← model(c, q̈)
    if not DTD(â, th_i, δ) then
        break
    end
    k ⟵ k + 1
end
```

4 Experiments

4.1 Dataset

The National Science R&D Technology Information Service (NTIS) center at the Korea Institute of Science &Technology Information (KISTI) is responsible for collecting all R&D project reports funded by the South Korean government, and over 0.6 M reports were collected as of 2019. From the entire collection, we selected 3,000 reports according to high TF-IDF scores after tokenization using KoNLPy[4]. Then, problems (PBs) and solutions (SLs) in the titles and abstracts of the reports were tagged by three annotators. We only used the titles and abstracts because they explicitly include problems and solutions because of their nature. Tab. 2 shows the token statistics of our dataset. The average length of reports is about 114 tokens, which is relatively short. The average lengths of problems and solutions are approximately about 7. According to Tab. 3, there are more problems than solutions in both the total and average. Reports have a problem at least but can have no solutions. The dataset was randomly split into training and test datasets with a 9:1 ratio. Tab. 4 shows the number of reports regarding the occurrences of problems and solutions. Severe data imbalance is observed as $(\mathrm{PB}=1, \mathrm{SL} \leq 1)$ dominates both training and test datasets, whereas there are a few $(\mathrm{PB} \geq 2, \mathrm{SL}=1)$. Tab. 5 shows the average occurrences of types of PB and SL.

Table 2: Statistics of tokens

	# of tokens			
	Avg	Min	Max	Std
Reports	114.76	63	253	23.25
Problems	7.35	2	55	4.27
Solutions	7.14	2	57	5.42

[4]https://konlpy.org/en/latest

Table 3: Statistics of annotations

	Total	Average	Min	Max
Problems	3,077	1.06	1	5
Solutions	2,384	0.82	0	6

Table 4: Occurrences report types

	# of docs				
	$PB = 1,\ SL \le 1$	$PB = 1,\ SL \ge 2$	$PB \ge 2,\ SL = 1$	$PB \ge 2, SL \ge 2$	Total
Train	2,531	74	43	52	2,700
Test	281	8	5	6	300
Total	2812	82	48	58	3,000

Table 5: Average occurrences of types

	# of types			
	PB = 1, SL >= 2	PB >= 2, SL = 1	PB >= 2, SL >–= 2	PB = 1, SL <= 1
PB	1.000	2.125	2.500	1.000
SL	3.200	1.000	2.334	0.800

4.2 Setup

We implemented all QA models using PyTorch[5] and performed the experiments using TITAN RTX ∗ 4 GPUs. The hyper-parameter settings, following the default setting of KoELECTRA, are summarized in Tab. 6. The maximum turn in DTD was set to 3, indicating that three problems or solutions can be extracted at maximum from a report. As an evaluation metric, we opted for a character-level F1 score computing the harmonic mean of the precision and recall over characters between the predicted answer and ground truth, which is used in QA tasks.

Table 6: Hyperparameter setup

Model hyper parameters	
Vocab size	32,200
# of layers	12
# of heads	12
Embedding size	768
Hidden size	768
Max sequence length	512

(Continued)

[5]https://pytorch.org/

Table 6: Continued

Model hyper parameters	
Learning rate	0.002
Lower case	False
Finetuning stage hyper parameters	
Optimizer	Adam
Training epochs	5
Warmup steps	0
Learning rate	0.0005 (0.001 in the 2nd phase)
Max Query length	64
Max Answer length	30
Max Sequence length	512
Candidate answer size	20
Batch size	16
Max turn	3

4.3 Results

4.3.1 Performance

Five QA models were evaluated in our experiments: a previous multi-turn model [18], a standard single-turn model, and three variants of our multi-turn model. We adopted the previous multi-turn model [18] with three modifications to our task: replacing BERT with ELECTRA, changing the question prompt to incorporate the history, and including three training phrases. Single-turn denotes a standard single-turn QA model. Multi-turn A utilizes DTD and QR using only the immediate history at each turn. Multi-turn B is similar to Multi-turn A but different in QR as it incorporates the full history to the question prompt at each turn. In Multi-turn C, few-shot learning for ($PB \geq 2,\ SL \geq 2$) was applied with Multi-turn B to alleviate the severe data imbalance. Tab. 7 shows the performance comparison of the five QA models. Surprisingly, the single-turn model produced the best performance, 78.79 and 76.01, on PB and SL, respectively. It reveals that QR and few-shot learning are effective as the best performance, 76.92 and 73.23 for PB and SL, respectively, among the four multi-turn models (baseline and three variants) obtained with Multi-turn C. However, it was lower than the single-turn model, which is not what we expected.

Table 7: Performance comparison of four QA models

	F1		
	PB	SL	ALL
Single-turn	**78.79**	**76.01**	**77.58**
(Li et al., 2019) [18]	72.25	68.77	70.35
Multi-turn A	76.00	71.59	74.01
Multi-turn B	76.02	72.29	74.34
Multi-turn C	**76.92**	**73.23**	**75.25**

The multi-turn models obtained lower performance when compared to the single-turn model; this can be explained using Tab. 8, which compares the performance of report types with respect to the occurrences of PBs and SLs. It shows that our multi-turn model, except for the previous research, has a tendency for strength when there are more targets of PBs and SLs. In $(\mathrm{PB} = 1, \mathrm{SL} \geq 2)$ and $(\mathrm{PB} \geq 2,\ \mathrm{SL} \geq 2)$ report types, our multi-turn model outperformed the single-turn model, whereas it produced slightly lower performance in $(\mathrm{PB} = 1, \mathrm{SL} \leq 1)$. However, the gap, -0.83, was reduced compared to that in Tab. 8. In $(\mathrm{PB} \geq 2, \mathrm{SL} = 1)$ report type, our multi-turn model was still degraded. We assumed that the multi-turn model was not properly trained because of insufficient training examples of 43 for $(\mathrm{PB} \geq 2, \mathrm{SL} = 1)$.

Table 8: Performance comparison according to report types

	F1			
	PB = 1, SL >= 2	PB >= 2, SL = 1	PB >= 2, SL >= 2	PB = 1, SL <= 1
Single-turn	31.20	**32.27**	36.04	**83.41**
(Li et al., 2019) [18]	21.74	24.59	24.20	79.44
Gain	−9.46	−7.68	−11.84	−3.97
Multi-turn C	**44.06**	28.76	**43.52**	82.58
Gain	**+12.86**	−3.51	**+7.48**	−0.83

4.3.2 Ablation Studies

We explored the effects of the turn-level TAV in DTD and the second phase of the training on the QA dataset. All ablation studies were performed in the same experimental setup as explained. To investigate the contribution of TAV in DTD, we conducted experiments with two variations based on Multi-turn C:

- **-TAV**: Multi-turn C without TAV (no threshold)
- **-turn-level thresholds:** Multi-turn C using TAV without turn-level thresholds (a fixed threshold).

Tab. 9 reveals the significant effect of TAV on DTD. Performance degraded without TAV on Multi-turn C because the downstream turns were not detected correctly in DTD. In addition, we observed that the turn-level thresholds are essential to use with TAV as the performance of TAV without the turn-level thresholds is worse than the one without TAV itself. In Tab. 10, we demonstrate the effectiveness of the second phase training on the QA dataset to accommodate QA representations. The performance in Single-turn and Multi-turn C with the second phrase training is improved compared to those without it.

Table 9: Performance comparison of four QA models

	F1				ACC
	PB = 1, SL >= 2	PB >= 2, SL = 1	PB >= 2, SL >= 2	PB = 1, SL <= 1	DTD
Single-turn	31.20	**32.27**	36.04	**83.41**	NA

(Continued)

Table 9: Continued

	F1				ACC
	PB = 1, SL >= 2	PB >= 2, SL = 1	PB >= 2, SL >= 2	PB = 1, SL <= 1	DTD
Multi-turn C	**44.06**	28.76	**43.52**	82.58	**93.27**
- turn-level thresholds	29.81 (−14.25)	19.74 (−9.02)	37.59 (−5.93)	82.58 (−0.00)	62.53 (−30.68)
- TAV	25.05 (−19.01)	20.42 (−8.34)	37.13 (−6.39)	82.58 (−0.00)	49.80 (−43.47)

Table 10: Evaluation results of the second phase

	F1
Single-turn	**77.58**
- 2nd phase training	75.77
Multi-turn C	75.25
- 2nd phase training	73.71

4.3.3 Error Analysis

The results extracted in the second turn using different QR strategies for problems and solutions are presented in Tab. 11. Cases 1 and 2 are for problems and solutions, respectively. In Case 1, Multi-turn without the history information produced inaccurate outputs, whereas the ones with the history information (Multi-turn A and B) produced an accurate result "…development of technology to manufacture fixed bodies (…고정체 제조 기술 개발)". Similarly, Multi-turn A and B produced more accurate results than Multi-turn without the history information in Case 2. The output "…using precise grinding processing (…정밀연삭 가공을 이용한)" of Multi-turn B is perfectly matched to the Ground Truth (GT). Furthermore, the inaccurate outputs of Multi-turn without the history information caused repeated extraction. In Cases 1 and 2, a single-turn model failed to produce the results because of a lack of dealing with multiple extractions.

Table 11: Outputs of proposed QA models

Model	Output	
	Case 1. Multiple problem extraction	**Case 2**. Multiple solution extraction
GT (k = 1)	"…development of localization manufacturing technology for implants" ("…임플란트 국산화 제조기술 개발")	"…using HIP and precision grinding processing" ("…HIP 및 정밀연삭 기술을 이용한")
GT (k = 2)	"…**development of technology to manufacture fixed bodies**" ("…**고정체 제조기술 개발**")	"**…using precise grinding processing**" ("… **정밀연삭가공을 이용한**")
	Q: "What is 2nd problem except '…' in context?	**Q**: "What is 2nd solution except '…' in the context?
Single-turn	NA	NA
Multi-turn w/o H	"…development of localization manufacturing technology for implants" ("…임플란트 국산화 제조기술 개발")	"…using HIP and precision grinding techinque" ("…HIP 및 정밀연삭 기술을 이용한")
Multi-turn A	"…development of technology to manufacture fixed bodies" ("…고정체 제조기술 개발")	"using implants…using precise grinding processing" ("임플란트…정밀연삭 가공을 이용한")
Multi-turn B	"**…development of technology t o manufacture fixed bodies**"	"**…using precise grinding processing**"

5 Conclusions and Future Work

This paper presents a multi-turn QA model, a mechanism to extract problems and solutions from Korean R&D reports. By the QR and DTD methodologies, multi questions and downstream turns are made, and we have effectively handled the target extraction as multi-turn QA. Our proposed model trained on the three-phase training procedures and can prevent the multiple and repeated extractions at the document level. Our model is trained through three phases with different purposes. A series of experiments on the Korean R&D report dataset we built showed the effectiveness of our model and the in-depth analysis of the results and behaviors. In our future work, we plan to construct more annotated data and extend our model to deal with the relationship between problem and solution.

Funding Statement: This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019R1G1A1003312) and the Ministry of Education (NRF-2021R1I1A3052815).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] B. Lent, R. Agrawal and R. Srikant, "Discovering trends in text databases," *KDD*, vol. 97, pp. 227–230, 1997.

[2] B. Yoon and Y. Park, "A text-mining-based patent network: Analytical tool for high-technology trend," *Journal of High Technology Management Research*, vol. 15, no. 1, pp. 37–50, 2004.

[3] Y. G. Kim, J. H. Suh and S. C. Park, "Visualization of patent analysis for emerging technology," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1804–1812, 2008.

[4] S. Lee, B. Yoon and Y. Park, "An approach to discovering new technology opportunities: Keyword-based patent map approach," *Technovation*, vol. 29, no. 6–7, pp. 481–497, 2009.

[5] M. J. Shih, D. R. Liu and M. L. Hsu, "Discovering competitive intelligence by mining changes in patent trends," *Expert Systems with Applications*, vol. 37, no. 4, pp. 2882–2890, 2010.

[6] D. Thorleuchter and D. V. Poel, "Web mining based extraction of problem solution ideas," *Expert Systems with Applications*, vol. 40, no. 10, pp. 3961–3969, 2013.

[7] Y. Zhang, X. Zhou, A. L. Porter and J. M. Vicente Gomila, "How to combine term clumping and technology roadmapping for newly emerging science & technology competitive intelligence: 'Problem & solution' pattern based semantic TRIZ tool and case study," *Scientometrics*, vol. 101, no. 2, pp. 1375–1389, 2014.

[8] Y. Zhang, D. K. R. Robinson, A. L. Porter, D. Zhu, G. Zhang *et al.,* "Technology roadmapping for competitive technical intelligence," *Technological Forecasting and Social Change*, vol. 110, pp. 175–186, 2016.

[9] X. Wang, Z. Wang, Y. Huang, Y. Liu, J. Zhang *et al.,* "Identifying R&D partners through subject-action-object semantic analysis in a problem & solution pattern," *Technology Analysis & Strategic Management*, vol. 29, no. 10, pp. 1167–1180, 2017.

[10] J. An, K. Kim, L. Mortara and S. Lee, "Deriving technology intelligence from patents: Preposition-based semantic analysis," *Journal of Informetrics*, vol. 12, no. 1, pp. 217–236, 2018.

[11] L. Qiu, H. Zhou, Y. Qu, W. Zhang, S. Rong *et al.,* "QA4IE: a question answering based framework for information extraction," in *The Semantic Web-ISWC 2018: 17th International Semantic Web Conference, Part 1*, 1st ed., vol. 11136 LNCS. Cham, Switzerland: Springer, pp. 198–216, 2018.

[12] O. Levy, M. Seo, E. Choi and L. Zettlemoyer, "Zero-shot relation extraction via reading comprehension," in *Proc. CoNLL 2017-21st Conf. on Computational Natural Language Learning*, Vancouver, Canada, pp. 333–342, 2017.

[13] B. McCann, N. S. Keskar, C. Xiong and R. Socher, "The natural language decathlon: Multitask learning as question answering," arXiv preprint arXiv:1806.08730, pp. 1–23, 2018.

[14] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, Minneapolis, MN, USA, vol. 1, pp. 4171–4186, 2019.

[15] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer *et al.,* "SpanBERT: Improving pre-training by representing and predicting spans," *Trans Assoc Comput Linguist*, vol. 8, pp. 64–77, 2019.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi *et al.,* "RoBERTa: A robustly optimized BERT pretraining approach," arXiv preprint arXiv:1907.11692, pp. 1–13, 2019.

[17] K. Clark, M. T. Luong, Q. V. Le and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," arXiv preprint arXiv:2003.10555, pp. 1–18, 2020.

[18] X. Li, F. Yin, X. Li, A. Yuan, D. Chai *et al.,* "Entity-relation extraction as multi-turn question answering," in *Proc. ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, Florence, Fl, Italy, pp. 1340–1350, 2019.

[19] D. Wadden, U. Wennberg, Y. Luan and H. Hajishirzi, "Entity, relation, and event extraction with contextualized span representations," in *Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and 9th Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, HK, China, pp. 5784–5789, 2019.

[20] T. Zhao, Z. Yan, Y. Cao and Z. Li, "Asking effective and diverse questions: A machine reading comprehension based framework for joint entity-relation extraction," in *Proc. of the Twenty-Ninth Int. Joint Conf. on Artificial Intelligence (IJCAI)*, Yokohama, Japan, pp. 3948–3954, 2020.

[21] Z. Chen and C. Guo, "A pattern-first pipeline approach for entity and relation extraction," *Neurocomputing*, vol. 494, pp. 182–191, 2022.

[22] S. Lim, M. Kim and J. Lee, "KorQuAD1.0: Korean QA dataset for machine reading comprehension," arXiv preprint arXiv:1909.07005, pp. 1–5, 2019.

[23] P. Rajpurkar, J. Zhang, K. Lopyrev and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proc. of the 2016 Conf. on Empirical Methods in Natural Language Processing, Proc. (EMNLP)*, Austin, TX, USA, pp. 2383–2392, 2016.

[24] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao *et al.,* "MS MARCO: A human generated machine reading comprehension dataset," in *Proc. CEUR Workshop Proc.*, Barcelona, Spain, vol. 1773, 2016.

[25] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni *et al.,* "NewsQA: A machine comprehension dataset," in *Proc. of the 2nd Workshop on Representation Learning for NLP*, Vancouver, Canada, pp. 191–200, 2017.

[26] M. Joshi, E. Choi, D. S. Weld and L. Zettlemoyer, "TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension," in *Proc. of the 2017 Conf. 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada, vol. 1, pp. 1601–1611, 2017.

[27] A. Fisch, A. Talmor, R. Jia, M. Seo, E. Choi *et al.,* "MRQA 2019 shared task: Evaluating generalization in reading comprehension," in *Proc. of the 2nd Workshop on Machine Reading for Question Answering*, Hong Kong, HK, China, pp. 1–13, 2019.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," in *Proc. of the 31st Int. Conf. on Neural Information Processing Systems (NIPS'17)*, Long Beach, California, USA, vol. 30, pp. 6000–6010, 2017.

[29] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan *et al.,* "Language models are few-shot learners," in *Proc. of the 33st Int. Conf. on Neural Information Processing Systems (NeurIPS'20)*, Virtual conference, vol. 33, pp. 1877–1901, 2020.