applied sciences

Article

Ensemble-NQG-T5: Ensemble Neural Question Generation Model Based on Text-to-Text Transfer Transformer

Myeong-Ha Hwang [1], Jikang Shin [1], Hojin Seo [1], Jeong-Seon Im [1], Hee Cho [1] and Chun-Kwon Lee [2,]*

[1] Digital Solution Laboratory, Korea Electric Power Research Institute (KEPRI), 105 Munji-ro, Yuseong-gu, Daejeon 34056, Republic of Korea
[2] Department of Control and Instrumentation Engineering, Pukyong National University, Busan 48513, Republic of Korea
* Correspondence: ck.lee@pknu.ac.kr

Abstract: Deep learning chatbot research and development is exploding recently to offer customers in numerous industries personalized services. However, human resources are used to create a learning dataset for a deep learning chatbot. In order to augment this, the idea of neural question generation (NQG) has evolved, although it has restrictions on how questions can be expressed in different ways and has a finite capacity for question generation. In this paper, we propose an ensemble-type NQG model based on the text-to-text transfer transformer (T5). Through the proposed model, the number of generated questions for each single NQG model can be greatly increased by considering the mutual similarity and the quality of the questions using the soft-voting method. For the training of the soft-voting algorithm, the evaluation score and mutual similarity score weights based on the context and the question–answer (QA) dataset are used as the threshold weight. Performance comparison results with existing T5-based NQG models using the SQuAD 2.0 dataset demonstrate the effectiveness of the proposed method for QG. The implementation of the proposed ensemble model is anticipated to span diverse industrial fields, including interactive chatbots, robotic process automation (RPA), and Internet of Things (IoT) services in the future.

Keywords: neural question generation; deep learning; natural language processing; ensemble algorithm

Citation: Hwang, M.-H.; Shin, J.; Seo, H.; Im, J.-S.; Cho, H.; Lee, C.-K. Ensemble-NQG-T5: Ensemble Neural Question Generation Model Based on Text-to-Text Transfer Transformer. *Appl. Sci.* **2023**, *13*, 903. https://doi.org/10.3390/app13020903

Academic Editor: Valentino Santucci

Received: 22 November 2022
Revised: 23 December 2022
Accepted: 4 January 2023
Published: 9 January 2023

1. Introduction

Recently, chatbots provide responses to a wide range of requests through communications based on automated rules and natural language processing (NLP). Accordingly, the application of the technology has indeed contributed to the development of areas such as table reservations, web browsing, and shopping [1,2]. In addition, various chatbot frameworks using deep learning have emerged, and relevant research is underway within multiple industries. However, difficulties yet exist in generating a question-and-answer (QA) training dataset for deep-learning-based chatbots, such as LUIS, Watson Conversation, API.ai, and RASA [3–6]. Moreover, the process is laborious and inconvenient, as the dataset is created manually. Question generation (QG) refers to the task of automatically generating questions from various inputs such as texts, databases, and semantic representations. Through QG, the model may generate questions apart from the QA pairs presented by the QA task, thereby creating an augmentation effect and demonstrating an overall improvement in QA performance [7,8]. As for the case of other NLP task research, neural question generation (NQG) research in the QG field is being conducted by generating questions according to the given context and providing answers using the Pretrained Language Model (PTM). PTMs, which refer to models that can learn language representations using a large-size corpus [9], possess the advantage of not requiring human resources to build a training dataset for the chatbot as the dataset can be produced automatically.

The initial NQG model mainly utilized the recurrent neural network (RNN). RNN is an artificial neural network in which the connections between each node possess a cyclic structure. Therefore, a key advantage of the RNN is the capacity for wide application within the NLP field due to specialization in sequence-based signal processing [10–13]. However, a critical drawback is a reduction in accuracy for texts comprising long sentences. Since a sequential input is utilized for the RNN-based NQG model, accuracy can decrease due to the vanishing gradient, which refers to the lessened influence on words located further apart. For the case of a convolutional neural network (CNN)-based NQG model, convolution filters function in extracting relevant information while scanning through data [11]. However, limitations exist, as longer texts require both the stacking of multiple convolution layers to learn words located near the end of the text and an amplified amount of calculation.

Alternatively, through the application of an attention mechanism that places emphasis on the order and correlation between words, the transformer supplements the drawbacks of CNNs and RNNs, which generally utilize simple sequences [14]. The transformer can be largely divided into two mechanisms: First is an encoding block that compresses information regarding correlations between every word within the input sentence and converts the information into a latent representation. Second is a decoding block that outputs the probability of occurrence for each word in consideration of both the information sent from the encoder (reference sentence structure) and the input from the decoder (input sentence). Recently, the focus has shifted towards improving QG prediction performance for each model through PTM, incorporating the basic encoder–decoder block of the transformer. Bidirectional encoder representations from transformer (BERT) is configured to enable attention in two directions rather than one by using an encoder block. BERT is highly suitable for extracting the meaning of sentences, because more information can be reflected during the encoding process [15]. Generative pretraining (GPT) uses only the decoder block in the transformer structure to learn to predict the next word when a new word is given [16].

Three types of NQG models developed through the fine-tuning of BERT, including BERT-QG, BERT-SQG (sequential QG), and BERT-HLSQG (high-light sequential QG), have been proposed by Ying-Hong Chen et al. [17]. Moreover, an NQG model incorporating a fine-tuned GPT-2 model was proposed by Luis Enrico Lopez et al. [18]. MixQG, a QG model that combines nine question answering datasets containing various answer types and applies them to T5 and BART, was proposed by Lidiya Murakhovs'ka et al. [19]. As a result, it was possible to generate queries with different cognitive levels under different answer type conditions, and about 10% performance improvement was demonstrated compared with the T5 and BART algorithms. To ensure the complexity and quality of the question, Zichu Fei et al. extracted key entities using a Graph Attention Network (GAT) and proposed CQG, a QG model that applied BERT and flag tags using the extracted key entities [20]. As a result of the HotpotQA performance experiment, it proved to have about 25% performance improvement compared with the existing model at 5 BLEU. Chenyang Lyu et al. proposed a QG model that uses an article to summarize, and it applies to the BART model [21]. The model was able to apply tasks simultaneously as a supervised model and an unsupervised model and demonstrated higher performance compared with the existing model. As such, it can be seen that research on transformer-based NQG models is being actively conducted.

Additionally, the ensemble strategy for solving the challenge of answering questions in natural language based on the attention mechanism emerged. To combine the question answering, Anna Aniol et al. employed the Bidirectional Attention Flow (BiDAF), QANet, and Mnemonic Reader models [22–25]. Following the acquisition of class-specific voting weights for each question word, this model is a technique of answering in which the candidate response with the highest weight is chosen from a pool of many candidates. As a result, contemporary research is being undertaken utilizing an ensemble model based on the attention mechanism.

Unlike BERT or GPT, which use either the encoder block or decoder block of the transformer but not both, the text-to-text transfer transformer (T5) model simultaneously uses the encoder block and the decoder block, hence being adequate for application to all NLP tasks. In addition, its recent use in various NLP fields has confirmed high performance in comparison to other models [26].

In this paper, we propose an algorithm to construct a dataset that generates a large number of questions of high accuracy to train a chatbot service using a T5-based ensemble model (Ensemble-NQG-T5). In specific, when a representative question is selected among questions extracted from multiple NQG models of high similarity, a bias occurs due to high redundancy while training the chatbot algorithm. The issue, however, can be overcome using the proposed method.

Utilization of the soft-voting method for questions generated between each NQG model ensures a high similarity with the input text, a low correlation between the generated questions, and concomitantly increases the number of generated questions, hence being suitable for constructing a PTM learning dataset. To do so, the average of the Bilingual Evaluation Understudy (BLEU) and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scores from each NQG model is selected as weights [27,28]. Then, candidate questions are generated for each model, and weight information calculated in advance for each model is assigned to the generated questions. Subsequently, the relevance of each question with other generated questions is calculated, and questions with a similarity exceeding the threshold are considered the same question. Whether to select a question is finally decided through a soft-voting method that adopts a question generated from a high-accuracy model according to the initial assigned weight. Consequently, compared with the single NQG method, the quantity and quality of questions can be enhanced.

The rest of the paper is organized as follows: Section 2 provides a review of the pretrained model (PTM) and evaluation metrics of the NQG algorithm. In Section 3, we present the theory and process of the proposed Ensgmble-NQG-T5 and present an experiment, comparing results and discussion in Section 4. We conclude this paper in Section 5.

2. Related Work

2.1. Pretrained Models

Various PTMs models have been developed through further optimization: BERT refers to a model specialized for classification by stacking the multiple encoder blocks of the transformer. XLNet applies the factorization order technique, which enables the consideration of all sequences. RoBERTa is an optimized model of BERT, such as training time, batch size, etc. MASS introduces a masking technique that is opposite to encoder–decoder. BART adds various noising methods to the encoder. Finally, MT-DNN enables universal representation by applying to multitask learning to BERT [15,29–33].

As a result, pretraining with a huge corpus and fine-tuning to downstream tasks have shown to significantly improve performance for a variety of activities. Existing PTMs, on the other hand, had limitations in that they were unable to complete all NLP tasks, including those involving question answering, summarizing, and machine translation. In addition, a T5 algorithm was developed, which defined all NLP tasks as “Text-to-Text” tasks. In various QA task performance evaluation experiments, such as GLUE, SuperGLUE, and SQuAD, T5 demonstrated high performance and accuracy [34–36]. The structure of the T5 model represents that of the typical transformer, however, performance is improved by suggesting several different learning methods. The first one is a word prediction method, which allows the prediction of words by denoising the noising input as a pretraining objective. Second, overfitting is prevented regardless of the size of the training data by using an unbalanced dataset for training. Third, the target data type is set to a text rather than a numerical value for universal applicable to all NLP tasks. Finally, state-of-the-art (SOTA) performance was demonstrated by training a model with 11 billion parameters on over 1 trillion tokens.

Figure 1 depicts the encoder–decoder structure of the transformer, which translates the input sentence. The transformer consists of an encoder block that compresses information regarding the reference sentence structure and, subsequently, sends the information to the decoder block, as well as a decoder block that translates an input sentence utilizing information received from the encoder block as a reference. The transformer may solve the bottleneck of the existing Seq2Seq model, and high performance has been demonstrated using contextual information as an attention mechanism [14].

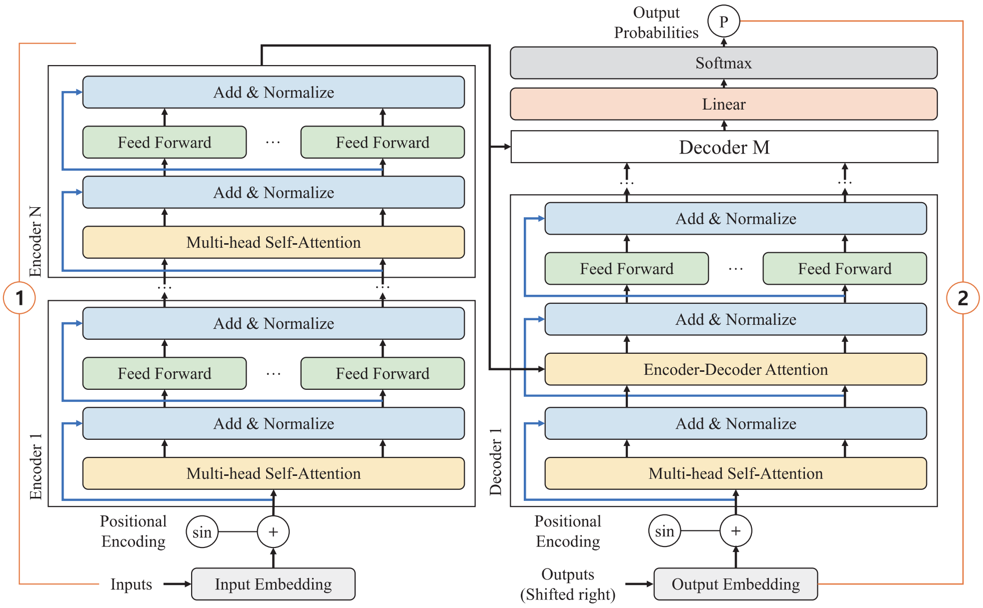

Figure 1. Architecture of text-to-text transfer transformer.

Through an embedding process and positional encoding, the input sentence is converted into vectors containing positional information for each word token. Embedded vectors consumed by the encoder undergo multiple self-attention processes as follows: First, each token generates three types of vectors, which include the representation (Query) of each word in the input sentence, Keys, which serve as labels for identifying relevant words, and the Value, which refers to the actual word representation. Second, the query in one token is multiplied by all the keys to calculate a score indicating significance, which then goes through a softmax to output the attention weight. The attention weights are finally multiplied by the values to output the relevance among words within the input sentence. To promote model learning, the input and output vectors are added as a residual connection, and the scores are normalized by layer normalization. Unlike the RNN, the accuracy of the output sequence does not decrease due to the vanishing gradient, since the entire input sentence is repetitively referred to.

The decoder block translates the input sentence or transforms the sentence structure by utilizing information from the encoder block regarding the relevance among words. The decoding block consists of two types of self-attention processes: the masked self-attention inside the decoding block and the attention between the output of the encoder and the decoder. Unlike the self-attention process, in which all word tokens within the input sentence of the encoding block are included, when word tokens including location information are sequentially input into the decoding block, only the attention score of the word tokens positioned upstream of each target word token is incorporated. In other words, as succeeding word tokens do not affect the self-attention process, predictions on successive words are made in the absence of information regarding words following the target word in the decoding process. For this process, information from the encoding process is used (encoder–decoder multihead attention): the Key and Value of the output of the encoder are utilized for determining the significance and correlation of words within the input sentence of the decoder block (encoder–decoder attention). Therefore, the calculation of the decoder attention score is affected by the output of attention within the encoder. Finally, output values from multiple decoder blocks go through the linear layer and softmax to output the final probability value of word order.

2.2. Evaluation Metrics of NQG

BLEU, an automatic evaluation method frequently used in machine translation tasks, determines the level of similarity of the translated candidate sentence with respect to the reference sentence. The presence of overlapping words and differences in the sentence length is compensated, and the geometric average is derived by changing the number of sequences (n-grams) of words of the candidate sentence that match with that of the reference sentence. BLEU-N can be expressed as follows [27]:

$$BLEU = BP * \exp(\sum_{n=1}^{N} w_n \log P_n) \tag{1}$$

Here, P_n refers to precision n-grams corrected for consecutive occurrences of the same word, and w_n refers to the weight for the precision of each gram. In this paper, overlapping sequences from grams 1 to 4 were calculated, and the weight was fixed at 1/4 each. In addition, BP is calculated as follows to provide a disadvantage when candidate sentences are either longer or shorter than the length of the reference sentence, which is the standard when translating.

$$BP = \begin{cases} 1 & \text{for} \quad c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{for} \quad c \leqq r \end{cases} \tag{2}$$

In addition, the ROUGE-L score used in this paper is a modified model of ROUGE. The method allows measurements of similarity using the Longest Common Subsequence (LCS) between candidates and references [28] and, as the F-score is obtained for the longest sequence in the candidate question, a greater value indicates higher accuracy in the word order of the translated sentence. Unlike other ROUGE score methods, implementing the LCS eliminates the requirement for consecutive matches. Moreover, as the longest in-sequence common n-gram can be automatically identified, the length of the n-gram need not be defined in advance. While BLEU focuses on precision, which refers to the frequency of n-grams in machine-generated translated questions occurring in the reference sentences, the ROUGE score functions in the measurement (recall) of the number of grams of the reference sentence that appears in the candidate sentence. Therefore, the two scores are complementary.

For assessment of similarity between two sentences, the reference sentence is defined as X, the length of X as m, the candidate sentence as Y, and the length of Y as n. Then, recall (R_{lcs}) and precision (R_{lcs}) are calculated in accordance with the formula below, and F_{lcs} is applied to the final formula of ROUGE-L. Parameter β plays functions in adjusting the relative significance of P_{lcs} and R_{lcs}. The ROUGE-L is calculated as follows:

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{3}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \tag{4}$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \tag{5}$$

3. Ensemble-NQG-T5

3.1. Overview

Figure 2 depicts the architecture of Ensemble-NQG-T5 for different fine-tuned models to NQG function-based T5. First, questions are generated for each NQG model using the SQuAD context dataset, and model evaluation scores for QG from the each model (BLEU, ROUGE) are calculated. Then, the accuracy weights of each question are calculated. Subsequently, candidate questions are generated for each model by using the target context for QG as input. Generated questions are assigned weight information calculated in advance for each generated model. Then, the similarity between questions is calculated in a round-robin manner. Questions possessing a similarity exceeding the threshold are considered identical, and questions generated from a model with higher accuracy are adopted according to preassigned weights. Subsequently, soft-voting classification is performed, where the remaining questions are discarded. Therefore, a question dataset may be generated for each purpose by optimizing the similarity between questions and the quantity of generation.

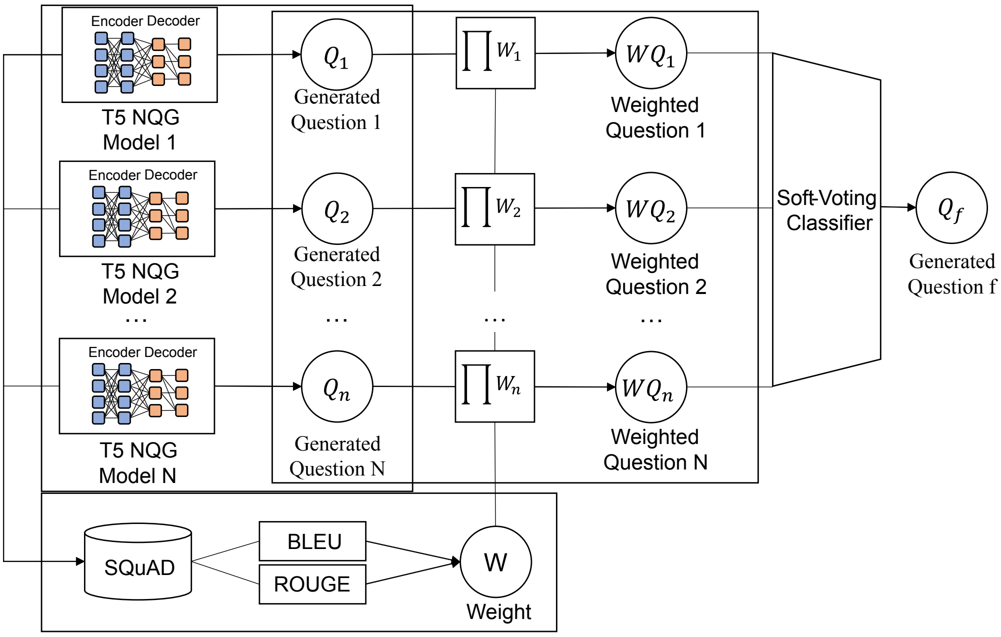

Figure 2. Architecture of Ensemble-NQG-T5.

First, each NQG model generating a question using the input context was fine-tuned to the output form of QG based on T5 [37,38]. During the fine-tuning process, distinct parameters are set and learned for each NQG model. Hence, the questions generated from each model are expressed differently, resulting in distinct QG performance [39]. Table 1 depicts the NQG parameters for each model. The NQG model of the Single Task type uses the Context (c) as an input, and a dictionary of QA pairs is generated as the output. Apart from output in the form of a dictionary, the multitask-type NQG model also includes a function to produce answers (a) as an output in response to inputs of question (q) and context pairs. Finally, the NQG model of the end-to-end type takes the Context as an input and generates only questions without answer supervision. Therefore, when comparing the quantity of QG using the same context, the end-to-end-type NQG model is most optimal. However, since questions are generated in the absence of supervision on the answer, the accuracy of the generated query is relatively low. Consequently, although a large amount of data to learn the algorithm can be guaranteed, the reliability of the chatbot's accuracy post-learning is low.

Table 1. The Neural Question Generation Models in Ensemble-NQG-T5.

Model Number	**1**	**2**	**3**	**4**	**5**	**6**
Type	Single Task	Multi Task	Single Task	Multi Task	End-to-End	
Task	T5 Small		T5 Base		T5 Small	T5 Base
Model Size	230.8 MB		850.3 MB		230.8 MB	850.3 MB

Therefore, in this study, an ensemble model (Ensemble-NQG-T5) based on different T5 models is presented, and through a soft-voting method between the datasets of the generated questions, the accuracy and quantity of QG are improved compared with those of the single-model-based NQG.

Detailed descriptions of Ensemble-NQG-T5, depicted in Figure 3, are as follows: First, SQuAD is used to generate model-specific questions from the input context for six pretrained and fine-tuned NQG-T5 models, each of different forms. Since model and task types differ, questions are created in the context of the same declarative text. Hence, even if the order of the words is fixed, a variety of question types with similar contents may be generated through the addition or deletion of words. For each model-specific question generated, BLEU and ROUGE scores are calculated in relationship with the target context, which is subsequently used as the weight of Ensemble-NQG-T5.

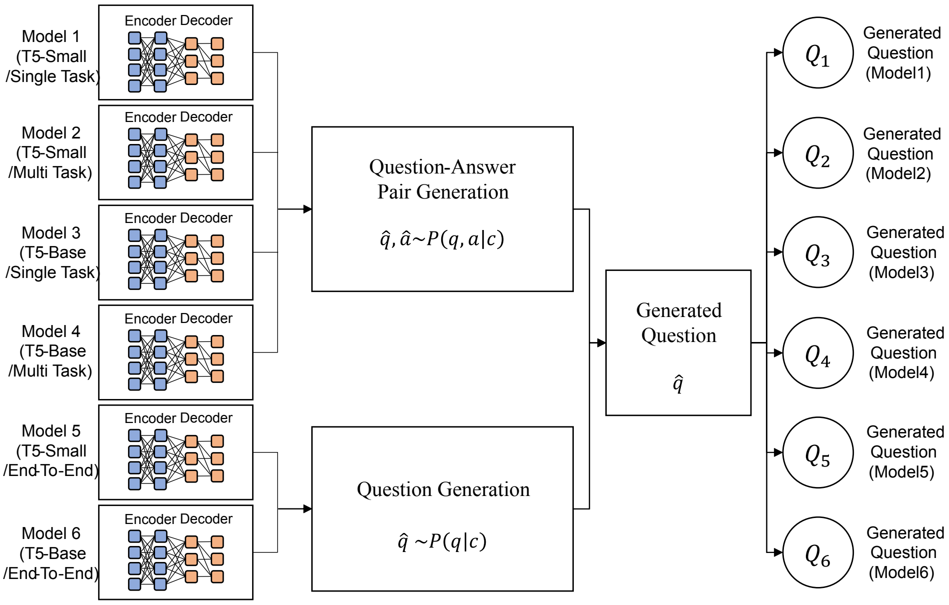
Model 1 (T5-Small /Single Task)
Model 2 (T5-Small /Multi Task)
Model 3 (T5-Base /Single Task)
Model 4 (T5-Base /Multi Task)
Model 5 (T5-Small /End-To-End)
Model 6 (T5-Base /End-To-End)
Encoder Decoder
Question-Answer Pair Generation
$\hat{q}, \hat{a} \sim P(q, a|c)$
Question Generation
$\hat{q} \sim P(q|c)$
Generated Question
$\hat{q}$
Q_1 Generated Question (Model1)
Q_2 Generated Question (Model2)
Q_3 Generated Question (Model3)
Q_4 Generated Question (Model4)
Q_5 Generated Question (Model5)
Q_6 Generated Question (Model6)

Figure 3. Process of Multi Question Generation in Ensemble-NQG-T5.

3.2. Soft-Voting Classifier

A higher BLEU is observed in response to a larger number of words included in the reference sentence, as the n-grams of the candidate sentence are set as the standard. A higher ROUGE can be obtained through an increased number of matches between n-grams in the candidate sentence, with the provided reference sentence as the standard.

For questions generated from Ensemble-NQG-T5, BLEU-N removes duplicate words and emphasizes the accuracy of machine translation made with the existing human dataset, and ROUGE-L places importance on word order in the question. Hence, both measurements are averaged and used as an indicator for the word order and accuracy of the translated questions for each model of Ensemble-NQG-T5.

The generated questions for each model are given a weight based on the BLEU and ROUGE scores, and when the similarity between the questions generated between the models is high, it becomes an index for calculating the selection priority.

Translated questions ultimately become different sentences depending on what the question is specifically asking for, or the answer type, regardless of identical use of words and word order. Therefore, interrogative sentence classification based on Semantic Role Labeling is implemented by first classifying questions generated for each model into 6 categories (i.e., Who, What, When, Where, and How). Then, the similarity is measured for each weighted question within each category, and when the Mutual Similarity Threshold (MST) is exceeded, clustering is performed so that questions can be grouped within the same cluster. The similarity measurement method utilizes cosine similarity [40]. Then, by comparing the weight values of each weighted question in each cluster, $WQ_{i,j}$ with the highest weight value is selected and designated as the Q_f. Algorithm 1 represents the pseudocode of the Soft-Voting Classifier for selecting the question (Q_f) of Ensemble-NQG-T5.

Algorithm 1 Soft-Voting Classifier of Ensemble-NQG-T5

```
procedure QUESTION GENERATION
    $n \leftarrow$ Model Number
    $m \leftarrow$ Final Number of Generated Questions
    $5W1H \leftarrow$ Class of What, Why, When, Who, Where, How
    $MST \leftarrow$ Mutual Similarity Threshold
    $WQ_{i,j} \leftarrow$ Weighted Question (i, j)
    $Q_f \leftarrow$ Selected Question
    for $i = 1, 2, \ldots, N$ do
        for $j = 1, 2, \ldots, M$ do
            Interrogative Sentence Classification based on Labeling of $WQ_{i,j}$
        end for
    end for
    for $x$ to $5W1H$ do
        for $y$ to # of each cluster do
            Clustering based on MST
        end for
        Weight Value Comparison for Each Cluster
        $Q_f = WQ_{i,j}$ with Best Weight Value
    end for
end procedure
```

4. Experimental Results and Discussion

4.1. Dataset and Experimental Setup

The SQuAD Version 2.0 was used for verification of the proposed algorithm [41]. The dataset includes SQuAD 1.1, which is a QA paired machine reading comprehension (MRC) dataset generated through cloud sourcing of a more than 500 Wikipedia article dataset. Over 100,000 QA pairs comprise the dataset, enabling wide usage as a standard benchmark dataset in relevant research fields. Therefore, the context and QA randomly extracted from SQuAD was used as the input, the similarity of output questions among other generated questions and the input sentence are calculated, and the final question is selected through evaluating weight values. Experiments within this study used a high-capacity RAM-based TPU from Google Colab, which includes 40 CPUs with Intel(R) Xeon(R) 2.30 GHz specification, a memory of 35.25 GB, and Ubuntu 18.04.5 LTS operating system. Moreover, Python 3.7 version was utilized.

4.2. Results

Table 2 depicts performance evaluation results of each model used for Ensemble-NQG-T5, including BLEU-N and ROUGE-L scores, along with the number of questions generated using the SQuAD 2.0. Results confirm that the BLEU-N and ROUGE-L scores of the T5-Base-based model were observed to be relatively higher than that of the T5-Small-based model due to differing pretraining sizes: For a single task, the BLEU-N score was higher by 0.029, while for the ROUGE-L score it was by 0.014. For multitask, the BLEU-N score was higher by 0.017, while for the ROUGE-L score it was by 0.014. Finally, for the end-to-end approach, the BLEU-N score was higher by 0.045, while the ROUGE-L score was by 0.018.

Table 2. Performance Evaluation for Each Single T5 Model.

Model Number	1	2	3	4	5	6
BLEU-N	0.447	0.465	0.476	**0.482**	0.426	0.471
ROUGE-L	0.422	0.427	0.436	**0.441**	0.407	0.425
# of GQ	767	772	778	807	**1117**	827

Regarding the quantity of QG, a larger number of questions were generated in the case of multitask. Specifically, for the T5-Small model, 5 more questions were generated in the case of multitask, and for the T5-Base model, 29 more questions were generated for multitask. Conversely, the end-to-end approach resulted in the generation of 290 more questions when incorporating the T5-Small model compared with the T5-Base model.

Table 3 describes results of ensemble optimization using questions generated from Ensemble-NQG-T5. For each category, the number of generated questions and the BLEU-N and ROUGE-L scores were compared when assigned different MST. At a low MST, even if the similarity between generated questions was low, only one representative question was extracted. Therefore, the resulting BLEU-N and ROUGE-L scores were high, at 0.556 and 0.595, respectively. However, only 415 questions could be extracted. On the other hand, at a high MST, only one representative question was extracted when the similarity between the questions was high. Hence, both the BLEU-N and ROUGE-L scores were observed to be low, at 0.462 and 0.427, respectively. Conversely, a large number of 5123 questions was derived.

Table 3. Performance Evaluation of Ensemble-NQG-T5 by Mutual Similarity Threshold.

Mutual Similarity Threshold	BLEU-N	ROUGE-L	# of GQ
0.1	**0.556**	0.595	415
0.2	0.555	**0.599**	617
0.3	0.546	0.552	963
0.4	0.517	0.505	1303
0.5	0.522	0.473	1708
0.6	0.517	0.445	2055
0.7	0.490	0.429	2480
0.8	0.472	0.421	2945
0.9	0.461	0.416	3458
1.0	0.462	0.427	**5123**

In other words, as a trade-off relationship exists between BLEU-N and ROUGE-L scores and the number of generated questions, specific results can be derived according to the user's needs by adjusting the MST.

Figure 4 depicts comparative results of evaluations on the Ensemble-NQG-T5 and single model results. For Ensemble-NQG-T5, the average evaluation score increased by up to 29.7 percent, and the amount of questions generated increased by up to 607.0% compared with the single T5 model results. Consequently, using this proposed algorithm, it is possible to create a question dataset for chatbots with higher quality and quantity than a single model, taking into account the characteristics of the amount of generated questions and mutual similarity.

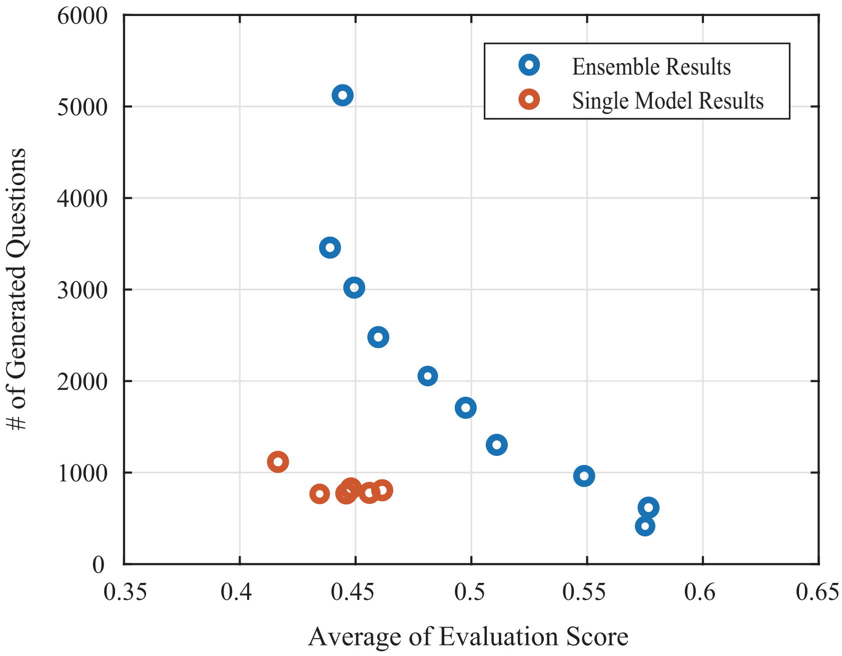

Figure 4. Comparison results of question generation.

5. Conclusions and Future Work

In this study, Ensemble-NQG-T5 was proposed to achieve an increased diversity and quantity of generated questions compared with the existing single NQG model. Ensemble-NQG-T5 is an ensemble-type NQG model based on T5, which allows compensation for the shortcomings of the existing NQG model by using a soft-voting classifier for the generated question dataset. The performance of QG from the context with the proposed algorithm was verified by a SQuAD QA dataset. The results reveal that generating questions using the ensemble-NQG-T5 algorithm can generate a larger number of nonoverlapping questions than using a single NQG model. It is possible to apply various up-to-date NQG evaluation methods to increase the number and quality of questions, such as QAScore, which scores generated questions without predefined reference and answerability, considering additional information of question type, as well as BERTscore, which is a performance measurement method using BERT [42–45]. Ensemble-NQG-T5 can be applied to improve natural language understanding (NLU) performance, since it helps to build a training dataset for deep-learning-based chatbots. In the future, the proposed algorithm will be extended to a platform linked to robotic process automation (RPA) and the Internet of Things (IoT) using the deep learning chatbot learned through Ensemble-NQG-T5.

Author Contributions: Conceptualization, M.-H.H. and J.S.; methodology, M.-H.H. and J.S.; software, M.-H.H., H.S. and H.C.; validation, M.-H.H.; formal analysis, M.-H.H.; resources, M.-H.H. and J.S.; data curation, M.-H.H. and J.-S.I.; writing—original draft preparation, M.-H.H. and C.-K.L.; visualization, M.-H.H.; supervision, C.-K.L. and J.S.; project administration, J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Korea Electric Power Corporation (KEPCO).

Conflicts of Interest: The authors declare no conflict of interest. The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Adamopoulou, E.; Moussiades, L. Chatbots: History, technology, and applications. *Mach. Learn. Appl.* **2020**, *2*, 1–18. [CrossRef]
2. Albayrak, N.; Ozdemir, A.; Zeydan, E. An overview of artificial intelligence based chatbots and an example chatbot application. In Proceedings of 2018 26th Signal Processing and Communications Applications Conference, Izmir, Turkey, 2–5 May 2018.
3. Microsoft. Available online: https://luis.ai/ (accessed on 21 November 2022).

4. IBM. Available online: https://www.ibm.com/watson/developercloud/conversation.html/ (accessed on 21 November 2022).
5. Google. Available online: https://www.api.ai/ (accessed on 21 November 2022).
6. Rasa Technologies Inc. Available online: https://rasa.com/ (accessed on 21 November 2022).
7. Pan, L.; Lei, W.; Chua, T.-S.; Kan, M.-Y. Recent Advances in Neural Question Generation. *arXiv* **2019**, arXiv:1905.08949.
8. Zhou, Q.; Yang, N.; Wei, F.; Tan, C.; Bao, H.; Zhou, M. Neural Question Generation from Text: A Preliminary Study. In Proceedings of the 6th National CCF Conference on Natural Language Processing and Chinese Computing, Dalian, China, 8–12 November 2017.
9. XiPeng, Q.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; Huang, X. Pre-trained models for natural language processing: A survey. *Sci. China Tech. Sci.* **2020**, *63*, 1872–1897.
10. Mulder, W.D.; Bethard, S.; Moens, M.-F. A survey on the application of recurrent neural networks to statistical language modeling. *Comput. Speech Lang.* **2015**, *30*, 61–98. [CrossRef]
11. Duan, N.; Tang, D.; Chen, P.; Zhou, M. Question Generation for Question Answering. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017.
12. Du, X.; Shao, J.; Cardie, C. Learning to Ask: Neural Question Generation for Reading Comprehension. *arXiv* **2017**, arXiv:1705.00106v1.
13. Wang, Z.; Lan, A.S.; Nie, W.; Waters, A.E.; Grimaldi, P.J.; Baraniuk, R.G. QG-Net: A Data-Driven Question Generation Model for Educational Content. In Proceedings of the Fifth Annual ACM Conference on Learning at Scale, London, UK, 26–28 June 2018.
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st Conference on Neural Information Process System (NIPS), Long Beach, CA, USA, 4–9 December 2017.
15. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the Conference NAACL-HLT: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019.
16. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. Available online: https://blog.openai.com/language-unsupervised (accessed on 21 November 2022).
17. Chan, Y.H.; Fan, Y.C. BERT for Question Generation. In Proceedings of the 12th International Conference on Natural Language Generation, Tokyo, Japan, 29 October–1 November 2019.
18. Lopez, L.E.; Cruz, D.K.; Cruz, J.C.B.; Cheng, C. Simplifying Paragraph-level Question Generation via Transformer Language Models. In Proceedings of the PRICAI 2021: Trends in Artificial Intelligence, Hanoi, Vietnam, 8–12 November 2021.
19. Murakhovs'ka, L.; Wu, C.-S.; Laban, P.; Niu, T.; Lin, W.; Xiang, C. MixQG: Neural Question Generation with Mixed Answer Types. In Proceedings of the Conference NAACL, Seattle, DC, USA, 10–15 July 2022.
20. Fei, Z.; Zhang, Q.; Gui, T.; Liang, D.; Wang, S.; Wu, W.; Huang, X. CQG: A Simple and Effective Controlled Generation Framework for Multi-hop Question Generation. In Proceedings of the 60th Annual Meeting of ACL, Dublin, Ireland, 22–27 May 2022.
21. Lyu, C.; Shang, L.; Grahan, Y.; Foster, J.; Jiang, X.; Liu, Q. Improving Unsupervised Question Answering via Summarization-Informed Question Generation. In Proceedings of the Conference EMNLP, Punta Cana, Dominican Republic, 7–11 November 2021.
22. Seo, M.; Kembhavi, A.; Farhadi, A.; Hajishirzi, H. Bi-Directional Attention Flow for Machine Comprehension. *arXiv* **2018**, arXiv:1611.01603.
23. Yu, A.W.; Dohan, D.; Luong, M.-T. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *arXiv* **2018**, arXiv:1804.09541.
24. Hu, M.; Peng, Y.; Huang, Z.; Qiu, X.; Wei, F.; Zhou, M. Reinforced Mnemonic Reader for Machine Reading Comprehension. *arXiv* **2018**, arXiv:1705.02798.
25. Aniol, A.; Pietron, M.; Duda, J. Ensemble approach for natural language question answering problem. In Proceedings of the 2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW), Nagasaki, Japan, 26–29 November 2019.
26. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
27. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002.
28. Lin, C.Y. ROUGE: A package for automatic evaluation of summaries. In Proceedings of the ACL-04 Workshop (Text Summarization Branches Out), Barcelona, Spain, 25–26 July 2004.
29. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized autoregressive pretraining for language understanding. In Proceedings of the Advances in Neural Information Processing System, Vancouver, BC, Canada, 8–14 December 2019.
30. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
31. Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T.-Y. MASS: Masked sequence to sequence pre-training for language generation. In Proceedings of the International Conference on Machine Learning(ICML), Long Beach, CA, USA, 10–15 June 2019.
32. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv* **2019**, arXiv:1910.13461.

33. Liu, X.; He, P.; Chen, W.; Gao, J. Multi-Task Deep Neural Networks for Natural Language Understanding. *arXiv* **2019**, arXiv:1901.11504.
34. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Bressels, Belgium, 31 October–4 November 2019.
35. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. SuperGLUE: A Sticker Benchmark for General-Purpose Language Understanding Systems. In Proceedings of the 33rd Conference Neural Information Processing System, Vancouver, BC, Canada, 8–14 December 2019.
36. Raipurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv* **2016**, arXiv:1606.05250.
37. Chan, Y.H.; Fan, Y.C. A Recurrent BERT-based Model for Question Generation. In Proceedings of the Second Workshop on Machine Reading for Question Answering, Hong Kong, China, 4 November 2019.
38. Lopez, L.E.; Cruz, D.K.C.; Cruz, J.C.B.; Cheng, C. Transformer-based End-to-End Question Generation. *arXiv* **2020,** arXiv:2005.01107.
39. Patil, S. Question Generation Using Transformers. Available online: https://github.com/patil-suraj/question_generation/ (accessed on 21 November 2022).
40. Singhal, A. Modern Information Retrieval: A Brief Overview. *IEEE Data Eng. Bull.* **2001**, *24*, 35–43.
41. Rajurkar, P.; Jia, R.; Liang, P. Know What You Don't Know: Unanswerable Questions for SQuAD. *arXiv* **2018**, arXiv:1806.03822.
42. Nema, Q.; Khapra, M.M. Towards a Better Metric for Evaluating Question Generation Systems. In Proceedings of the Conference EMNLP, Brussels, Belgium, 31 October–4 November 2018.
43. Ji, T.; Lyu, C.; Jones, G.; Zhou, L.; Graham, Y. QAScore-An Unsupervised Unreferenced Metric for the Question Generation Evaluation. *Entropy* **2021**, *24*, 1154. [CrossRef] [PubMed]
44. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. In Proceedings of the ICLR, Virtual Event, 27–30 April 2020.
45. Xu, W.; Napoles, C.; Pavlick, E.; Chen, Q.; Callison-Burch, C. Optimizing Statistical Machine Translation for Text Simplification. *Trans. ACL* **2016**, *4*, 401–415. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.