

# Semantic-Based Self-Critical Training for Question Generation

**Loïc Kwate Dassi**

Computer Science, Grenoble Institute of Technology, Ensimag, Grenoble, France

**Email address:**

Loic.Kwate-Dassi@grenoble-inp.org

**To cite this article:**

Loïc Kwate Dassi. Semantic-Based Self-Critical Training for Question Generation. *American Journal of Information Science and Technology*. Vol. 5, No. 4, 2021, pp. 93-97. doi: 10.11648/j.ajist.20210504.12

**Received:** September 17, 2021; **Accepted:** October 13, 2021; **Published:** November 17, 2021

---

**Abstract:** Question generation is a conditioned language generation task that consists in generating a context-aware question given a context and the targeted answer. Train language modelling with a mere likelihood maximization has been widely used while suffering from exposure bias and the discordance between the training and the test metrics. In the way of addressing this issue, The presented work portrays a fully Transformer-based reinforcement learning generator-evaluation architecture for neural question generation. To edge the flexibility of the generation, a semantic-based reward score was externally infused during the training to drive the training of the language model. The global architecture is laid out in a generator-evaluator fashion optimized directly to n-gram and semantic-based metrics. Evaluation metrics for language modelling only based on n-gram overlapping do not consider semantic relations between reference and candidate sequences. To improve the evaluation step, a two-fold evaluation was carried out. On the one side, an n-gram overlapping evaluation using the BLEU score. On the other side, a semantic-based assessment using BERTScore and NUBIA. The results were corroborated by a binary human evaluation of the semantic relatedness of the generated question and the ground truth. The results obtained showed that use a semantic-based REINFORCE algorithm for the question generation syntactically reshapes the generated questions while preserving their underlying semantic meaning. Many downstream applications can be drawn from a successful question generation including the enlargement of question answering datasets, the improvement of conversational systems, the enhancement of autonomous educational assessment systems, and so forth.

**Keywords:** Question Generation, Transformer, Reinforcement Learning, Generator-Evaluator, Language Generation

---

## 1. Introduction

Transformer architecture [1] has broken new ground in Natural Language Processing. Nowadays, it is widely used in a broad range of tasks including language understanding, language generation [2, 3] as well as computer vision [4]. The question generation falls into the category of sequence-to-sequence task, thanks to the ability of Transformer architecture with positional encoding to approximate any continuous sequence-to-sequence transformations within compact support [5]. Question generation is a sensitive task because it requires first a good understanding of the context, then requires how to establish a relation between the context and answer, and finally requires the model the ability to generate the questions fluently as a human. Through this work, an investigation of the potential of Transformer-based models on a question generation task is undertaken, and

similar results close to the state-of-the-art have been achieved while keeping the model smaller. More precisely, the architecture holds 220M trainable parameters as opposed to the large architecture of Prophenet [21] leading the state-of-the-art with 330M of trainable parameters. The neural networks developed in natural language processing, broadly in deep learning, become larger, these large models are energy demanding and are not widely accessible to the broad community. Investigating the aptness of the relatively small architectures to perform at the same stage of the larger architecture is noteworthy in the way of keeping the computational cost relatively small while pushing the boundaries of the state-of-the-art. This work is an extension of the work on Reinforcement Learning for question generation [6] where the generator was trained through the REINFORCE algorithm. In the generator-evaluator architecture implemented, the generator is a sequence-to-

sequence model Google's T5 [3] and the evaluator is the mixture of BLEU score [7] and the cosine similarity of the classification token's learning representation output by the Transformer-based model ELECTRA-discriminator [8]. The main long-term motivation of this work is the willingness to improve the education system by providing an autonomous system able to generate questions as humans, which can then be used to alleviate the conception of students' examinations and support self-taught students in their studies.

## 2. Related Work

Rule-based methods for question generation follow the directives set up by a specific expert. This range of models entirely relies on the knowledge of the expert who designs the rules of generating questions and the performance is limited by the expert's knowledge. Automatic Factual Question Generation from Text was introduced by [9]. Specifically focused on generating factual WH-questions, the designed system produces questions by successively applying a set of manually encoded transformation rules to the statements.

Qingyu Zhou et al. [10] introduced an RNN-GRU based encoder-decoder architecture that generates answer-aware questions. The encoder takes as inputs the context, the answer position, and linguistics features including part-of-speech (POS) and named entity recognition (NER). The decoder takes the conditioned learned representation produced by the encoder then generates context questions whose answers are spanned in the context.

Experiments on question-answer extraction were conducted by Patrick Lewis et al. [11]. They introduced Probably Asked Question, a large semi-structured knowledge database of about 65M pairs. The QA database was built using a question generation model and corpora from Wikipedia. To ensure the worthiness of the generated pair, they introduced a global filtering algorithm that ranks the generated pairs. Such a huge QA database could be useful to extend research on question generation and question answering tasks.

Question Generation by Transformers [12] explores the potential of the vanilla architecture of Transformer to deal with generating questions. Instead of fully encoding the context and answer as they appear in the dataset, some transformations including the change of the named entities by their corresponding NER tags were applied both on the context and answer.

A Reinforcement Learning Based Graph-to-Sequence Model was introduced by [6] showed a tailored use of reinforcement learning techniques (policy gradient algorithm) for language modelling. They introduced a reinforcement learning-based graph-to-sequence model for question generation, in turn, bringing in a deep alignment technique to improve the expressiveness of the encoder.

## 3. Model

Question generation can be identified as sequence

transduction consisting in transforming the combination of the context and answer sequences into question. In the setting, the context, the answer, and the question are sequences of words respectively identified by  $\{x_1^c, \dots, x_n^c\}$ ,  $\{x_1^a, \dots, x_m^a\}$ , and  $\{y_1, \dots, y_r\}$ . Here  $n, m$ , and  $r$  appropriately denoted the length of the context, the answer, and the question. Generate answer-aware questions consists in generating a question that maximizes the probability  $P(y_1, \dots, y_r | X^c, X^a)$ .

Previous related work focused on generating questions whose answers are spanned in the context; The extension of this limit was realised by simply modifying the structure of the model's inputs such that any answer could be attached to a given context rather than only the context spanned sections. An exhaustive test of this extension was not carried out in this work. The model's inputs are formed by the concatenation of context sequence and answer sequence separated by a special token  $[[CLS], x_1^c, \dots, x_n^c, [SEP], x_1^a, \dots, x_m^a]$ . Transformers are universal approximators of sequence to sequence transformations, thanks to the positional encoding and self-attention mechanism, the Transformer encoder will learn the positions of the answer, learn how to establish the relation with the context through the attention mechanism, and finally outputs a better contextual representation of the inputs. The input structure is also affordable to generate open-ended questions.

Drawing inspiration from reinforcement learning-based graph-to-sequence architecture [6], a fully Transformer-based reinforcement learning generator evaluator architecture is introduced to tackle question generation problems. Transfer learning has become an almost de facto technique in NLP like in Computer Vision, thanks to the massive and rich content data available on the internet like C4 (Colossal Clean Crawled Corpus), large models like GPT, BERT, T5 [13, 2, 3] can be trained through unsupervised or self-supervised learning with a large amount of unlabeled data. More general-purpose pre-training achieves better performance at a fine-tuning step on a downstream task like summarization, text classification, question answering, and language modelling.

T5 was used as the generator. Training language models that merely maximize the likelihood of the output sequence suffers from exposure bias and inconsistency in the training and evaluation performance measurement. This drawback is addressed in this work by using an evaluator that consists of the mixture of the metrics used at the validation step while optimizing the likelihood of the output sequence, the parameters of the models were also optimized according to metrics used at the evaluation time. The evaluator takes as inputs the generated question and the ground truth then outputs a reward score fed into the generator at the training time. To ensure the correctness of the generated question the mixture of BLEU and ELECTRA-discriminator was used as the evaluator, BLEU is used to increase the n-gram overlap between the ground truth and the predicted question, whilst ELECTRA is used for semantic relation. ELECTRA is a Transformer-based model trained with replaced token detection, a self-supervised learning task in which the model

learns how to distinguish whether or not a token is corrupted in the input sequence. ELECTRA converges much faster than its predecessor BERT and achieves better performance than BERT on downstream tasks, which points to its ability to produce a better contextual representation of the input sequences. The exploitation of this advantage leads to the building of a semantic evaluator. The reward function used in the adversarial training is defined as follows:

$$r_1(\hat{y}, y) = BLEU(\hat{y}, y) \quad (1)$$

$$r_2(\hat{y}, y) = \cos(E_m(\hat{y}), E_m(y)) \quad (2)$$

$$\tilde{r}(\hat{y}, y) = \alpha \times r_1(\hat{y}, y) + (1 - \alpha) \times r_2(\hat{y}, y) \quad (3)$$

$$r(\hat{y}, y) = \frac{\tilde{r}(\hat{y}, y) + 1 - \alpha}{2 - \alpha} \quad (4)$$

$\alpha$  is a scalar,  $y$  and  $\hat{y}$  designed the ground truth and the generated question (using greedy decoding) respectively,  $E_m(y)$  is the classification token's ([CLS]) vector representation provided by ELECTRA's encoder,  $r(\hat{y}, y)$  refers to the reward score.

The loss function used to optimize the model is a mixture of the basic likelihood of the output sequence and a modified likelihood which is called here the reinforcement learning loss. The basic component of the loss is merely the log-likelihood of the output sequence:

$$L_{base}(\hat{y}, y) = -\sum_t \log P(y_t | y_{<t}) \quad (5)$$

The added reinforcement learning loss function is defined as follows:

$$L_{rl}(\hat{y}, y) = -(1 - r(\hat{y}, y)) \sum_t \log P(y_t | y_{<t}) \quad (6)$$

The overall loss function is defined as a linear combination of the two-loss function components aforementioned:

$$L(\hat{y}, y) = \gamma \times L_{base}(\hat{y}, y) + (1 - \gamma) \times L_{rl}(\hat{y}, y) \quad (7)$$

$\gamma$  is a scalar. The reinforcement learning component of the overall loss function behaves as a self-critical part. It tends to lower the objective function when the generated question is semantically close to the ground truth. The use of self-critical training for image captioning [14] is fully justified because it significantly increases the performance of the test metrics on the MSCOCO task. The generation of insightful questions remains an open research question; the self-critical training paradigm provides one way to address this task. The generation of clarification questions was approached by Sudha Rao and Hal Daumé III. [15] with GAN-like architecture where the generator was a sequence-to-sequence model and discriminator a utility function to rank the generated question. The higher the rank, the more the answer to the question does not lie in the context provided to the generator.

## 4. Experiments

The training and the inference of the developed architectures were conducted with the benchmark reading

comprehension dataset SQUAD v1.1 [22], which consists of 100,000+ (context, question, answer) triple posed by crowdworkers on a set of Wikipedia articles. The data is originally split into two parts: the training and the test. The test set remained unchanged in the experiments and the training set was randomly split into a training and development set with the factors 94% and 6% respectively.

At the training time, the parameters of the evaluator remained frozen. The hyper-parameter searching was performed on the following hyper-parameters:  $\gamma, \alpha, learning\ rate\ (lr), batch\ size\ (bs)$ . The ranges used for these parameters were:  $0.05 \leq \alpha, \gamma \leq 1$ ;  $10^{-6} \leq lr \leq 10^{-4}$ ;  $bs \in \{16, 32, 64, 128\}$ . The scaling of the batch size was carried out via gradient accumulation technique. The best model regarding the loss was obtained with the following hyper-parameters:  $bs = 32$ ,  $lr = 1.17 \times 10^{-5}$ ,  $\alpha = 0.179$ ,  $\gamma = 0.09$ .

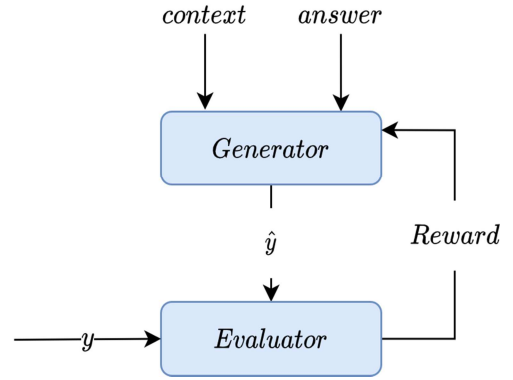


Figure 1. Model architecture.

## 5. Results

Following the previous works, the n-gram-based metric BLEU was used to evaluate the syntactic reconstruction ability of the models. Besides the n-gram based metrics, the semantic side of the generated questions was assessed with the use of semantic-based metrics, namely BERTScore [23], NUBIA [24]. The semantic relation and logical agreement scores were drawn from NUBIA. The evaluation was performed solely on the test set provided in the original dataset. Table 1 summarizes the evaluation for the metric BLEU and table 2 refers to the evaluation for the metric NUBIA. To ensure that the semantic-based adversarial training effectively improves the performance of the model toward the chosen metric, the following models were independently trained (1)  $T5_{base} + RL_{base}$  (BLEU), (2)  $T5_{base} + RL_{base}$  (BLEU + Semantic).

Table 1. Evaluation on the test set on the BLEU metric.

Model	BLEU	Model	BLEU
Transformer [1]	3.09	ASAs2s [18]	16.17
SeqCopyNet [17]	13.02	G2S <sub>sta</sub> +BERT+RL [6]	18.30
NQ++ [10]	13.29	T5 <sub>base</sub>	21.31
MPGR+R* [19]	14.71	T5 <sub>base</sub> + RL <sub>base</sub> (BLEU + Semantic)	21.52
AFQA [20]	15.64	T5 <sub>base</sub> + RL <sub>base</sub> (BLEU)	22.05
s2sa-at-mp-gsa [16]	15.82	ProphetNet [21]	23.91

## 6. Discussion

The first fact that could be drawn from the results of table 2 is the difference of the BLEU score of the two model configurations that have been undertaken namely (i) BLEU-based discriminator and (ii) the blending of BLEU and the semantic component to form the discriminator. Indeed the configuration (i) got a BLEU score greater than the configuration (ii). Tapping into how the discriminators of both structures have been made, this first result conveys that using the model is effectively trained toward improving the performance according to the metric used in the discriminator. That is, the BLEU score obtained with the discriminator only based on BLEU is greater than the score of the discriminator using the mixing of BLEU and the semantic similarity checker.

The second insight is drawn from the analysis of the semantic score of the two configurations.

Albeit the BLEU score of the configuration (ii) is less than the BLEU score of the configuration (i), the semantic-based scores of the configuration (ii) are commensurable with or even better than the semantic-based scores of the configuration (i). This reveals a change in the syntactic structure of the generated question while the semantic relatedness between the ground truth remains unchanged or even improved. This second fact corroborates the aforesaid hypothesis stating that the model's parameters are optimized toward enhancing the performance regarding the metric used in the discriminator component. This observation can be justified by the fact that the values of the hyper-parameters,

namely  $\alpha, \gamma$ , convey that the embedding cosine similarity of the representation outputs by the discriminator has a high contribution to the reward score and the reinforcement learning loss component has a high contribution to the overall loss.

Both architecture developed outperforms the vanilla architecture of the T5 base, this authenticates the usefulness of the generator-evaluator architecture set to address the question generation. In terms of BLEU score. The exploration of the large architectures of T5 was not carried out in this work. But instead, the potential of relatively small architecture to perform on par with the larger ones leading the leaderboard.

In addition to automatic evaluation, a binary human-based evaluation of the semantic correlation between the ground truth and the prediction was carried out. 500 questions were randomly sampled from the initial dataset (SQuAD v1.1) and the model used was the T5base + RLbase(BLEU + Semantic) configuration. The generated questions were classified in different WH-question classes, then for each class, the proportion of questions positively ranked by 5 crowdworkers was calculated. Based on the results in table 3, it is noted that the model hardly generates questions related to explanation whilst it has an affordable performance when the questions are related to time and date. The aim of this human evaluation step is to track the type of questions where the model has a good/bad performance to design a specific discriminator for each type of question in further work.

**Table 2.** Semantic evaluation on the test set. Semantic relation and Logical Agreement drawn from NUBIA.

Model	BLEU	BERTScore	Semantic Relation	Logical Agreement
T5 <sub>base</sub> + RL <sub>base</sub> (BLEU + Semantic)	21.52	52.60	61.51	44.16
T5 <sub>base</sub> + RL <sub>base</sub> (BLEU)	22.05	52.65	61.31	42.52

**Table 3.** Human evaluation and semantic evaluation on T5<sub>base</sub> + RL<sub>base</sub>(BLEU + Semantic).

Type	Frequency	Human Score	BERTScore	Semantic Relation	Logical Agreement
What/ Which	44.16%	67.92	50.55	64.80	48.63
Why/ How	5.83%	57.14	48.83	57.95	28.91
Where	4.16%	80	46.75	53	13.07
When	12.49%	100	64.31	74.8	61.88
Who	19.16%	69.56	59.56	63	45.75
How many	14.16%	76.47	53.95	57.2	29.65

## 7. Conclusion

This work presented a fully Transformer-based reinforcement learning Generator-Evaluator architecture (wherein the generator and the evaluator were off-the-shelf Transformer-based models) to address the question generation task. Toward improving the flexibility of the generated questions, a regularizer has been folded in the loss function to optimize the model to a specific metric. The results showed that the ablation study using only BLEU as the optimization metric outperforms the combination of

BLEU and semantic-based optimization on the BLEU metric meanwhile, the compounding of the BLEU and semantic-based optimization has a commensurable performance with the BLEU optimization on the BLEU score but performs better on semantic-based metrics. These results authenticate the use of the regularizer to optimize text generation to a given evaluation metric. A potential outlook of this work is the adaptation of the regularizer to a specific type of question, that is, models could be drawn to address specific kinds of questions: Remembering, Understanding, Applying, Analyzing, Evaluating, and Creating.

## References

- [1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
- [2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
- [3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.
- [4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.
- [5] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions?, 2020.
- [6] Yu Chen, Lingfei Wu, and Mohammed J. Zaki. Reinforcement learning-based graph-to-sequence model for natural question generation, 2020.
- [7] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. 2002. URL <https://www.aclweb.org/anthology/P02-1040.pdf>
- [8] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators, 2020.
- [9] Michael Heilman. Automatic factual question generation from text. 2015. URL [https://liti.cs.cmu.edu/sites/default/files/research/thesis/2011/michael\\_heilman\\_automatic\\_factual\\_question\\_generation\\_for\\_reading\\_assessment.pdf](https://liti.cs.cmu.edu/sites/default/files/research/thesis/2011/michael_heilman_automatic_factual_question_generation_for_reading_assessment.pdf).
- [10] Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. Neural question generation from text: A preliminary study, 2017.
- [11] Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. Paq: 65 million probably-asked questions and what you can do with them, 2021.
- [12] Kettip Kriangchaivech and Artit Wangperawong. Question generation by transformers, 2019.
- [13] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2020. URL [https://cdn.openai.com/research-covers/language-unsupervised/language\\_understanding\\_paper.pdf](https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf).
- [14] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning, 2017.
- [15] Sudha Rao and Hal Daumé III. Answer-based adversarial training for generating clarification questions, 2019.
- [16] Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. Paragraph-level neural question generation with max out pointer and gated self-attention networks. 2018. URL <https://www.aclweb.org/anthology/D18-1424.pdf>.
- [17] Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. Sequential copying networks, 2018.
- [18] Bang Liu, Mingjun Zhao, Di Niu, Kunfeng Lai, Yancheng He, Haojie Wei, and Yu Xu. Learning to generate questions by learning what not to generate. The World Wide Web Conference on - WWW '19, 2019. doi:10.1145/3308558.3313737. URL <http://dx.doi.org/10.1145/3308558.3313737>.
- [19] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. A graph-to-sequence model for amr-to-text generation, 2018.
- [20] Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He and Yanjun Ma, and Shi Wang. Answer-focused and position-aware neural question generation. 2018. URL <https://www.aclweb.org/anthology/D18-1427.pdf>.
- [21] Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training, 2020.
- [22] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.
- [23] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.
- [24] Hassan Kane, Muhammed Yusuf Kocyigit, Ali Abdalla, Pelkins Ajanoh, and Mohamed Coulibali. Nubia: Neural-based interchangeability assessor for text generation, 2020.