# Automatic Question Generation for Educational Questions

**Aakash Patel** [*1] **Narayanan Venugopal** [*1] **Rachel Himmel** [*2] **Shreyas Chandrashekaran** [*1] **Varun Goyal** [*3]

## Abstract

Assessment is integral to learning because it is designed to help individuals identify their weak points and enhance their knowledge to help them progress. Creating the right kind of questions to aid in assessment can be a difficult task, both for instructors and students alike. Automatic question generation from learning resources and course content would help alleviate this labor, but existing question generation models often perform well only on data where the answer is contained in the given context, generating questions directly based off of key terms in the material. These questions often do not reflect actual examination questions, especially in higher-level settings, which call for questions that require deeper thought and may not have a specific answer. To address this problem, we investigate the use of answer-agnostic question generation methods, including masked language models (MLM) and graph-based models (Graph2Seq), with a focus on educational questions. Our baseline model performs quite poorly at this task thus far, but that makes sense as it is not necessarily optimized for this unique and specific task.

## 1. Introduction

The earliest evidences of conducting assessments- whether oral or written- to evaluate one's knowledge have been found in the 13th century. Much clearer investigations suggest written examinations were not introduced until the 1700s [1]. Since then, the human race has been conducting examinations by devising questions and answers from texts and other resources. With copious information and data from over thousands of years, it has become imperative to be able to consolidate them and ask the right questions to the students to progressively test their knowledge and help them grasp better. While our teachers work hard to design each question to fulfil that purpose, it would be highly efficient and conducive to them if they had a help- particularly from an automated system that generates questions, is quick and precise. With the advancement in technology and programming languages, it is now possible to execute this idea. The teachers can potentially spend more time teaching and preparing their lectures best suited for their students, and not worry about generating questions from the material for knowledge assessment. From the opposite perspective, such an automated system will also be instrumental to students in self-learning by knowing what are the right questions to ask from a specific topic and think appropriately in that direction.

The majority of early attempts at question generation (QG) for educational applications involved creating "fill in the blank" questions from textbooks [2]. While this method has been proven to be fairly effective in classroom settings, there are many situations where these fill-in-the-blank questions are not effective in assessment, as they only address distinct key terms extracted directly from the material. At a higher academic level, especially in disciplines in the liberal arts, exam questions are more likely to become more abstract, lacking definitive answers. In this setting, generating questions directly from context material using answers that can be extracted from that material, i.e. answer-aware QG, often fails to produce useful or representative examination questions that can be used by instructors or as a study resource.

However, many existing models that perform question generation focus only on answer-aware QG, often conducted on the Stanford Question Answering Dataset (SQuAD) [3]. SQuAD is a reading comprehension dataset, consisting of questions posed on a set of Wikipedia articles, where the answer to every question is a span extracted directly from the corresponding passage. Though SQuAD is the most frequently used dataset for QG, its questions are often based on simple facts, such as "In what country is Normandy located?" and "Which city is the fifth-largest city in California?"

[*] Equal contribution [1] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, U.S.A. [2] Department of Statistics, University of Michigan, Ann Arbor, U.S.A. [3] Department of Mechanical Engineering, University of Michigan, Ann Arbor, U.S.A.. Correspondence to: Aakash Patel <aakashdp@umich.edu>, Narayanan Venugopal <narayanv@umich.edu>, Rachel Himmel <rhimmel@umich.edu>, Shreyas Chandrashekaran <shreyasc@umich.edu>, Varun Goyal <varungo@umich.edu>.

As our goal is to generate questions which can be applicable in an educational setting, we instead elected to use the LearningQ dataset [4], which consists of instructor-designed questions collected from TED-Ed talks along with learner-generated questions and audio transcripts collected from Khan Academy videos. For example, a sample question from LearningQ is “If you were given oxygen and hydrogen at the same temperature and pressure, which has more energy?” With the intent of generating higher-level questions, often without predetermined answers in mind, LearningQ is more suitable as the questions are more open-ended and less contrived. We plan on utilizing the LearningQ data to produce answer-agnostic question generation models (which do not require manual selection of answer spans) to produce questions that more closely align with advanced examination questions, potentially enabling QG that involves a greater synthesis of different aspects of the material and require deeper thought.

Additionally, these answer-agnostic models trained on the LearningQ data, specifically vis-à-vis the Khan Academy video transcripts, could conceivably prove useful in the context of producing study questions directly from lecture transcripts. In this era of Zoom course recordings, where students can have access to the transcripts of their classes, question generation models fine-tuned on educational transcripts could provide a valuable resource.

In this project, we will study answer-agnostic automatic question generation for educational methods, developing and comparing different approaches to the task, including transformer models, graph-based methods, and summarization models.

## 2. Previous Work

While there has been much research and literature produced with this goal in mind, there are certainly areas for improvement in this domain despite the extensive work done already. Surveys of existing literature in question generation indicate that there is little focus in the current literature on controlling the difficulty of generated questions or enhancing question forms and structures. Additionally, it has been a challenge to generate syntactically and semantically valid questions [5].

One of the main limitations of existing works is the simplicity of generated questions. Most existing work for generated questions consists of a few terms, or only produces some form of quick response question (simple multiple choice, fill in the blank, basic paraphrasing for short answer, etc.), which is less applicable at a higher academic level. Even open-ended questions frequently consist of a short (less than one sentence) answer that comes directly from the material. While these questions are still useful, especially in scenarios when the course material and assessment tends toward rote memorization of terms and content, there is definitely a potential for improvement by exploring the generation of more complex types of questions.

Huang and He in 2016 [6] developed a model introducing Lexical Functional Grammar (LFG) [7] as the linguistic framework for question generation, which enables systematic utilization of semantic and syntactic information. They optimized the structure using LFG and utilized semantic dependencies to select sentence from the text and paraphrase for QG. In another earlier work by Kunichika *et al.* [8], they generated five types of questions from a story text by extracting syntactic and semantic information (consisting of time and space information and information about verbs, nouns and modifiers) using Definite Clause Grammar (DCG). They also used dictionaries of synonyms and antonyms to generate questions related to finding the word-meanings or their opposites. However, these models relied largely on hand-crafted texts and descriptions that required specific domain knowledge consuming a lot of time.

In more recent research, the focus has shifted towards the use of neural networks for QG. These models are more generalizable, less time consuming, more efficient, data-driven, and provide end-to-end trainable framework. The first neural network based QG model was introduced by Du *et al.* [9]. Their model did not rely on hand-crafted rules and was end-to-end trainable using sequence-to-sequence (Seq2Seq) learning. In their model, they adopted the global attention mechanism to focus on crucial parts of the input for word generation during decoding, and optimize the conditional probability using RNN encoder-decoder architecture. Their model outperformed all rule-based models. Since then, different research groups have focused on improving QG using a similar approach by additionally introducing information about question types or answer position features. Some groups have modified the framework for encoding to enhance the efficiency of QG [5, 10], while other groups propose using different objective functions for optimization [11, 12, 13].

The use of large pre-trained language models has recently gained popularity across all NLP tasks, including question generation. Chan and Fan [14] demonstrate that pre-trained BERT language models can achieve high performance on the SQuAD question generation task. Similar to BERT, another transformer-based method is using the Text-to-Text Transfer Transformer (T5) model [15], which is based on the standard transformer architecture, but trained on a curated version of the Common Crawl web archive data with a variety of different tasks, such as classification, summarization, and question answering [16].

One limitation of such models is that they do not capture rich structure information in the text other than the word sequence itself. To resolve this, Chen *et al.* (2020) [5] proposed a reinforcement learning (RL) based graph-to-sequence (Graph2Seq) model for question generation. The model consists of a Graph2Seq generator with a Bidirectional Gated Graph Neural Network based encoder. A hybrid evaluator combining both cross-entropy and RL losses was used to ensure that questions were syntactically and semantically valid. Their model outperformed existing methods at the time on the standard SQuAD benchmark.

These models discussed above typically perform answer-aware question generation, with a majority of the work on QG models having been done using the SQuAD dataset. Because SQuAD contains answer spans extracted directly from the source material, and these answer spans are consistently very short, consisting of just a key term, much of the work that has been done does not address generating questions of a greater difficulty or higher level of abstraction. Existing QG methods mostly produce narrowly focused questions that have close syntactic relations to their associated answer spans, producing fact-based questions peripheral to the underlying document topic. These questions are likely not useful to a professor seeking to test student's understanding rather than memorization, or a student wishing to study material more deeply via practice questions. We intend to present a new perspective on question generation for this educational purpose.

Chen *et al.* argue that answer information is crucial for generating high quality questions from a passage, and the availability of an answer is integral to their model. However, when an answer is simply not available or not possible, we believe that a graph-based approach can still achieve good performance due to its ability to capture important structural information about the text. To this end, we plan on adapting this approach to be answer-agnostic and just use the context and the question as inputs when training the model.

## 3. Methods

### 3.1. Dataset

The Stanford Question Answering Dataset (SQuAD) [3] is the most frequently used dataset for machine reading comprehension tasks, including question generation. However, the questions in SQuAD are generally too simplistic and factual for a higher-level educational setting, so we experiment with the LearningQ dataset instead [4], which focuses on specifically curated educational questions. LearningQ consists of Khan Academy articles and video transcripts with viewer questions from the comments, as well as Ted-Ed video transcripts with questions written by domain experts. Compared to SQuAD, a much higher proportion of the LearningQ questions require multiple sentences or external information to answer (90.57% LearningQ vs 1.47% SQuAD) [4], justifying our choice in using LearningQ for more conceptual and open-ended questions.

There are almost 10,000 unique documents in LearningQ comprising a total of 223,858 questions, of which a vast majority (over 200k) are from the Khan Academy split. Because of this, we follow Chen *et al.* [4] and train our models mostly on the Khan Academy data, but also use the Ted-Ed data as a separate test split to assess performance. Each sample in the LearningQ data consists of a question, the entire source document from which it was drawn, and a limited selection of sentences from the source that are most relevant to the question. Due to computational limitations in processing large amounts of text, we use the limited selection for training most of our models, except for the summarization model which will be explained in more detail in Section 3.2.3.

10% of the data is reserved for testing and a further 10% is used for validation.

### 3.2. Model

#### 3.2.1. T5 Model

As a baseline, we first finetune a T5 transformer model to generate questions similar to those in the LearningQ data. We use a variant of T5 from HuggingFace, initially finetuned for question generation on the SQuAD dataset, as a base, and finetune the model further on the LearningQ data. The raw text data is first preprocessed, which consists of tokenization, truncation to at most 512 tokens, and max-length padding. The context passage is the input to the model, and the question is the output.

The model is finetuned for 7 epochs, with a batch size of 4, learning rate of $1 \times 10^{-4}$, and a gradient accumulation step size of 16.

#### 3.2.2. Graph2Seq Model

The Graph2Seq model proposed by Chen *et al.* [5] consists of three main components: a deep alignment network to align the answer with the passage, a bidirectional gated graph neural network (BiGGNN) with an RNN decoder, and a hybrid evaluator with a mixed objective function involving both cross-entropy loss and RL loss. We plan to implement a simplified version of this model, due to both a different task objective as well as time constraints.

The first point of difference is the deep alignment network. Since our dataset does not contain answers to the questions, and many of the questions are such that the answer could not readily be found in the passage, the alignment module is not well-suited for our objective. Therefore, we plan to bypass it entirely, and just construct the graph on the passage embedding alone. If time permits, we would like

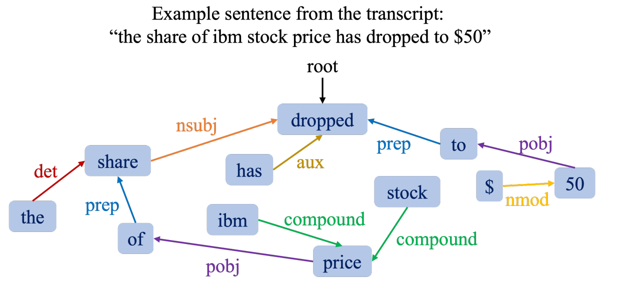

*Figure 1.* An example of conversion of a sentence from the LearningQ dataset to graph using dependency parsing for the Graph2Seq model generator. The link definitions can be found in [18].

to experiment with performing alignment with the question instead of the answer, since semantic relationships between the question and passage may also prove useful in generating higher quality questions.

In the paper, Chen *et al.* experiment with two different kinds of graphs, a syntax-based "static" graph built using dependency parsing (see Figure 1 for an example), and a semantics-aware "dynamic" graph that uses self-attention to compute an adjacency matrix on the aligned passage-answer embeddings. We plan to only work with the static syntax-based graph, mainly because the Graph2Seq model empirically performs better with the static graph.

Finally, our GNN, decoder, and evaluator components will be similar to the Chen *et al.* model. The GNN first learns the node embeddings from input passage, and then constructs the overall graph embedding based on the node embeddings.

At a high level, the node embeddings are computed as follows. At each node, the initial value is set to the word embedding vector of the corresponding word from the passage. Then for a desired number of steps, the embeddings from the outgoing edges ("forward neighbors") are aggregated via some choice of aggregation function (Chen *et al.* use a mean aggregate for the syntax-based graph), and the same for the incoming edges ("backward neighbors"). The aggregated representations are then fused and a GRU is used to update the current embedding. We refer readers to [5] and [17] for more in-depth mathematical details of bi-directional graph networks.

The graph embedding is then used by the RNN decoder to generate an output sequence, which is compared to the ground-truth questions to determine the loss. A generalized schematic outlining the major components of our Graph2Seq model appears in Figure 2.

### 3.2.3. Summarizer + GPT

Since its release in the past year, ChatGPT has become hugely popular for a variety of language tasks, such as text generation, question answering, translation, and summarization. To assess how well GPT performs on educational question generation, we plan to use GPT-3 [19] to generate questions based on the LearningQ data.

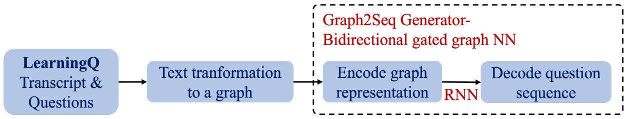

*Figure 2.* A schematic of the Graph2Seq model implementation for QG on the LearningQ dataset.

The context passages in the LearningQ data are considerably longer than those in SQuAD and other datasets. To limit inference time, we plan to first use a summarizer model to create a summary of the text data. Here, we will be using BERT to first summarize the data for the LearningQ dataset, using Miller's [20] approach to text summarization. The BERT summarizer will first be fine-tuned on the LearningQ dataset to improve the quality of the summaries. Following this, we will use the GPT API to generate questions from the summaries.

Since GPT is intended to be effective even in few-shot settings, we plan to just use the base GPT model without finetuning. We intend this approach to demonstrate the effectiveness of GPT for this task (or potential lack thereof), and so will leave further optimizations as future work.

### 3.2.4. Evaluation Metrics

In line with other works involving question generation, we are currently using the BLEU metric to evaluate our results [21]. However, as BLEU was designed for machine translation, it does not always generalize well to generative tasks such as question generation, and may not accurately measure a "good" question. We plan to address this by also evaluating our results using the Q-BLEU metric [22], which accounts for not only n-gram similarity but also answerability of generated questions.

## 4. Results

To this point, we have conducted one set of T5 experiments, and implementation details are in the methods section. Our first portion of this experiment was to finetune the already-finetuned T5 model on our LearningQ dataset, and we primarily focused on the Khan Academy dataset due to both its relative size and the more challenging nature of its questions. Unfortunately, the results were not promising on this dataset. Due to the sharp dichotomy between the outside-the-box nature of many of the questions used as labels in this portion of the data and the much simpler questions asked in the SQuAD dataset, the model seemed to suffer from relatedness issues with its questions, often querying about nonsensical phrases taken from the context. While the questions were generally grammatically-sounding questions, many of them were unintelligible given the context of the passage, and occasionally were just a sentence or phrase from the context with a question appended to it.

This makes sense when one considers the data patterns that the model was exposed to, especially that it was fine-tuned on SQuAD data before LearningQ data, with significantly different question patterns in both datasets. To try and find evidence for this hypothesis, we wanted to explore how T5 finetuned on the SQuAD data performed in a zero-shot transfer learning setting, so we ran an experiment where we strictly conducted inference using the Khan Academy test data. For that experiment, we received better numerical results; questions, however, resembled the simplistic style used in the SQuAD dataset and not the reasoning-based questions of our labels, resulting in an maximum average BLEU-1 of 0.137, BLEU-2 of 0.051, BLEU-3 of 0.016, and BLEU-4 of 0.002 when compared to reference questions. Table 1 below shows example sentences from both datasets.

This supported our earlier finding that the SQuAD dataset does not have similar questions in terms of complexity to those of LearningQ; therefore, as a side experiment on T5, we plan to finetune base T5 on only our datasets to see whether this aids in the type of question generation we are interested in. In addition to this, we plan to run all of these experiments on the TED-ED data, which seems more promising due to the more standard and curated nature of that data.

## 5. Future Milestones

Up to this point, we have been working on formulating our approach, general preprocessing of the data, and finetuning the baseline T5 model. We will direct our attention to the Graph2Seq model for most of the remainder of the project, as this is the most complex model and most time-consuming to implement part of our approach. The GPT + summarization will most likely not take too much time. We plan to reserve about a week at the end to work on the poster and the final report. A weekly breakdown of our tentative goals appears below:

- 3/10-17: Modify TED-ED data and run finetuning in the same way as Khan Academy, evaluate results with Q-BLEU and other metrics
- 3/17-24: Construct graph representation of dataset for application to Graph2Seq model, modification of Graph2Seq model for answer agnostic approach
- 3/24-31: Preliminary training and evaluation of Graph2Seq
- 3/31-4/7: Summarization model implementation
- 4/7-14: Graph2Seq ablation study, generate GPT questions from summaries
- 4/14-21: Create poster and write report

## 6. Author Contribution

A.P, S.C., and V.G. were involved in preprocessing the data, and S.C. handled writing and running the scripts for finetuning the T5 model. A.P. has begun initial work on constructing dependency graphs for the data. N.V., R.H., and V.G. conducted a thorough literature review to guide the problem formulation and methodology. All group members were involved in developing the overall plans and writing the report.

A tabular summary of author contributions is presented in Table 2.

## References

[1] S.-y. Teng, “Chinese influence on the western examination system: I. introduction,” *Harvard Journal of Asiatic Studies*, vol. 7, no. 4, pp. 267–312, 1943.

[2] M. Agarwal and P. Mannem, “Automatic gap-fill question generation from text books,” in *BEA@ACL*, 2011.

[3] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for machine comprehension of text,” *arXiv preprint arXiv:1606.05250*, 2016.

[4] G. Chen, J. Yang, C. Hauff, and G.-J. Houben, “Learningq: a large-scale dataset for educational question generation,” in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 12, 2018.

[5] Y. Chen, L. Wu, and M. J. Zaki, “Reinforcement learning based graph-to-sequence model for natural question generation,” *arXiv preprint arXiv:1908.04942*, 2019.

[6] Y. Huang and L. He, “Automatic generation of short answer questions for reading comprehension assessment,” *Natural Language Engineering*, vol. 22, no. 3, pp. 457–489, 2016.

[7] J. Bresnan, R. M. Kaplan, *et al.*, “Introduction: Grammars as mental representations of language,” *The mental representation of grammatical relations*, pp. xvii–lii, 1982.

| **Finetuned on SQuAD + Khan Academy** | **Finetuned only on SQuAD** |
|---|---|
| What did turner capture the small-like feeling of the speed of the train coming toward us? | What is a potter? |
| What is the name of the professor who's a professor here at stanford professor? | What is the name of Bell Laboratories? |
| What is the end date of 1700? | If 67 is in the array it is prime so what is the array? |

*Table 1.* Sample results from T5 + SQuAD and T5 + SQuAD + Khan Academy finetuning

| Author | Methodology | Data Processing | T5 model | Graph2Seq | Literature Review | Report |
|---|---|---|---|---|---|---|
| Aakash Patel | ✓ | ✓ | | ✓ | | ✓ |
| Narayanan Venugopal | ✓ | | | | ✓ | ✓ |
| Rachel Himmel | ✓ | | | | ✓ | ✓ |
| Shreyas Chandrashekaran | ✓ | ✓ | ✓ | | | ✓ |
| Varun Goyal | ✓ | ✓ | | | ✓ | ✓ |

*Table 2.* Summary of author contributions.

[8] H. Kunichika, T. Katayama, T. Hirashima, and A. Takeuchi, "Automated question generation methods for intelligent english learning systems and its evaluation," in *Proc. of ICCE*, vol. 670, 2004.

[9] X. Du, J. Shao, and C. Cardie, "Learning to ask: Neural question generation for reading comprehension," *arXiv preprint arXiv:1705.00106*, 2017.

[10] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," *Advances in neural information processing systems*, vol. 32, 2019.

[11] S. Johansen, K. Juselius, *et al.*, "Maximum likelihood estimation and inference on cointegration–with applications to the demand for money," *Oxford Bulletin of Economics and statistics*, vol. 52, no. 2, pp. 169–210, 1990.

[12] Z. Fan, Z. Wei, S. Wang, Y. Liu, and X.-J. Huang, "A reinforcement learning framework for natural question generation using bi-discriminators," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1763–1774, 2018.

[13] Y.-H. Liao and J.-L. Koh, "Question generation through transfer learning," in *Trends in Artificial Intelligence Theory and Applications. Artificial Intelligence Practices: 33rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2020, Kitakyushu, Japan, September 22-25, 2020, Proceedings*, pp. 3–17, Springer, 2020.

[14] Y.-H. Chan and Y.-C. Fan, "A recurrent bert-based model for question generation," in *Proceedings of the 2nd workshop on machine reading for question answering*, pp. 154–162, 2019.

[15] A. Roberts and C. Raffel, "Exploring transfer learning with t5: the text-to-text transfer transformer," *Google AI Blog*, 2020.

[16] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[17] K. Xu, L. Wu, Z. Wang, Y. Feng, M. Witbrock, and V. Sheinin, "Graph2seq: Graph to sequence learning with attention-based neural networks," *arXiv preprint arXiv:1804.00823*, 2018.

[18] M.-C. De Marneffe and C. D. Manning, "Stanford typed dependencies manual," tech. rep., Technical report, Stanford University, 2008.

[19] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[20] D. Miller, "Leveraging bert for extractive text summarization on lectures," 2019.

[21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the*

*Association for Computational Linguistics*, pp. 311–318, 2002.

[22] P. Nema and M. M. Khapra, "Towards a better metric for evaluating question generation systems," *arXiv preprint arXiv:1808.10192*, 2018.