Question Generation for Generating Textbook Flash Cards

Anonymous ACL submission

Abstract

One of the most effective ways of retaining the meaning of important concepts in learning materials is to review them in spaced intervals. Millions of students around the world are trying to do exactly that with the help of flash cards. In this short paper, we present a new, transformer-based application for education that automates the process of creating flash cards by automatically generating questions and answers for textbooks. As a proof-of-concept, we report two studies: a) generating questions for textbook summaries written by humans and b) testing a fully-automated pipeline. Several aspects of the quality of the resulting question-answer pairs are evaluated by three annotators. Finally, we describe and make available for review the deployed prototype for the flash card application.

1 Introduction

There is ample evidence from research in neuroscience and pedagogy that one of the most effective ways of learning hard concepts and making learning stick is the strategy of retrieving key ideas at spaced intervals (Oakley et al. (2018), Antony et al. (2017), Brown et al. (2014)). Flash cards are commonly used by students as an aid for this kind of intensive but effective learning strategy. A recent survey showed that 55% of college students already use flash cards as a study strategy (Miyatsu et al. (2018)). Research in this area has repeatedly shown that spaced repetition improves student performance significantly by helping them strengthen their comprehension of the materials and monitor their own learning. (Golding et al. (2012); Escobar Ibarra and Wong Martillo (2017); Wissman et al. (2012))

In this work, we build an intelligent tool that can identify concepts to be learned and generate

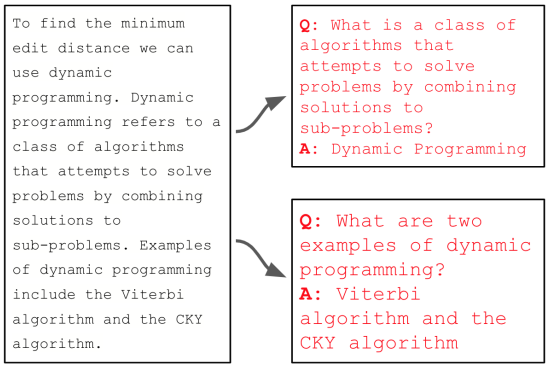

Figure 1: Questions automatically generated from learning material can be used as flashcards

question-answer pairs in the form of digital flash cards. Such a tool will have a substantial impact on improving learning outcomes and will reduce the effort required to generate flash cards manually. While continuing our research in addressing challenges in automating the generation of fluent questions and correct answers, we have deployed a fully functional prototype of such a tool which will be made available to the public for testing.

In Section (1), we report a proof-of-concept study on generating question-answer pairs on human and automated summaries and show that we can generate high quality flash cards as evaluated by three computer science teaching assistants. In section (4), we summarize our insights from analyzing errors and inter-annotator agreement. In Section (5), we present a short description of the deployed prototype of a flash card generation tool that will be made public after the anonymity period.

Our work makes the following contributions in the space of QG in education:

- Our pedagogical focus is on using QG to help learners retain the content that they consume by facilitating spaced learning activities such

as creating and reviewing flash cards on important concepts in their readings.

- We experiment with fine-tuning a pre-trained transformer model (Vaswani et al., 2017) to generate questions on summaries that we generate for textbook chapters.
- We conduct a human evaluation of several aspects of the quality of the generated question-answer pairs.
- We build a web-based prototype for automatically generating flash cards for educational content.

2 Prior Work

Automatic Question Generation (QG) is a popular research area that led to several applications in the educational domain. As (Kurdi et al., 2019) notes, the majority of the applications have focused primarily in the language learning domain and more recently expanded to other domains such as history and biology. Pedagogically, the primary focus of prior work was on using QG as an automated assessment tool, primarily for testing reading comprehension or vocabulary learning. Earlier work automated the generation of open cloze questions by omitting a word or phrase from a sentence (Pino and Eskenazi, 2009), generating distractors for cloze questions (Agarwal and Mannem, 2011; Narendra et al., 2013; Correia et al., 2012), and generating subjective questions using predefined templates or question patterns (Majumder and Saha, 2014; Bhatia et al., 2013). Methodologically, a range of question datasets were used (e.g., SQuAD, TriviaQA, NewsQA, RACE, LearningQ (Chen et al., 2018)) to train and evaluate QG models often with features extracted from preprocessing steps (e.g., shallow parsing and semantic distance). More recent work, also, focused on QG for assessment, including quizzes and formative questions and new tools were introduced to generate questions for online textbooks[1], history textbooks Pannu et al. (2018) and online class settings Zavala and Mendoza (2018).

Recent work in QG has clearly shown that large language models, when fine-tuned on QA datasets such as SQuAD (Rajpurkar et al., 2018), achieve very good performance on publicly available benchmarks (Lan et al., 2020), (Zhang et al., 2020), (Yang et al., 2020). While there are alternatives to this basic structure, such as using reinforcement learning (Chen et al., 2019), taking advantage of external knowledge graphs (Wang et al., 2020), creating synthetic data (Alberti et al., 2019), or developing novel pre-training tasks for few-shot learning (Ram et al., 2021), we decided to follow the majority and not experiment with such optimizations.

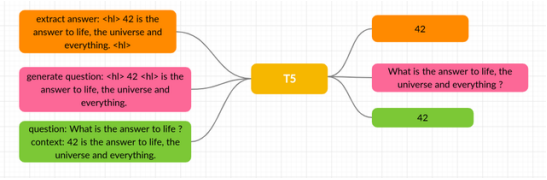

Figure 2: Diagram of the QA-QG model's three different fine-tuning tasks: Answer extraction, question generation, and question answering

3 Proof of Concept Experiments

As reviewed in the previous section, prior work on QG in the education domain focused on automating the process of generating cloze questions by selecting words for removal which the students were then asked to provide. Transferring close test models to the generation of questions given a text, generating flash cards for a textbook is not a trivial task. To evaluate the feasibility of generating flash cards for textbooks, we asked three teaching assistants of a Natural Language Processing course to write summaries for three chapters of the assigned textbook: Jurafsky and Martin (2009)'s textbook[2]. This step allows us to evaluate the performance of the QG model given a text that is clean of extraction errors and contains the most important concepts of the chapter. We then repeated the same experiment, this time replacing the human summaries with automated summaries. In both cases, we asked human annotators to evaluate the quality of the questions and the answers in the two conditions.

3.1 Method and Data

We follow the standard approach by using a fine-tuned language model, specifically a fine-tuned T5-base (60M parameters) model (Raffel et al., 2020), that is publicly available on the Huggingface ModelHub (Wolf et al., 2020).[3] This model was fine-tuned on SQuAD (Rajpurkar et al., 2018) to do

[1] https://get.vitalsource.com/what-we-offer/smart-course-generator

[2] https://web.stanford.edu/ jurafsky/slp3/

[3] https://huggingface.co/valhalla/t5-base-qa-qg-hl

three tasks: answer extraction, question generation, and question answering as shown in Figure 2. We use this model to extract answer-like spans of text (one per sentence) from a paragraph and then apply it again with a new task prefix to generate questions that would be likely to have those answers.[4]

For the first study, three human annotators (teaching assistants) wrote summaries for a total 123 sections of the Jurafsky and Martin (2009) textbook, chapters 2-4. We generated a total of 592 questions from these human-written summaries.

For the second study, we added a pre-processing step to fully automate the process. In order to capture the most important concepts of the content, we added an automatic summary generation step by once again using a large language model, this time a fine-tuned BART model Lewis et al. (2019)[5].

The results of both studies were evaluated by three annotators and we report them in the next section.

3.2 Evaluation

Out of the 592 generated question-answer pairs, we randomly selected 100 pairs for evaluation. We recruited three annotators, all computer science students, to evaluate the quality of the question-answer pairs. To gain insight into the possible aspects of error, we asked the annotators to answering the following questions: a) Is the question well-formed, b) is it relevant, c) is the answer correct and d) is the answer partially correct? The first two questions helps us distinguish errors due to failures in the linguistic expression of the question from errors due to errors due to the model's failure to ask a conceptually relevant questions.

In Table 1, we report the results of the evaluation by the three annotators. The results for the quality of questions generated for the human summaries are satisfactory, providing evidence that QG is a viable method for generating flash cards our of coherent summaries of textbook chapters. Not surprisingly, the performance for well-formedness on automated summaries. Not surprisingly, the worst performance for all metrics was for the questions generated in raw text. We take these results as evidence that indeed summaries include important concepts that need to be learned and retained. Overall, while these results are on low sample size (n=100) and the inter-annotator agreement is low, as we will see shortly, they indicate that generating questions on chapter summaries (whether human or automated) increases the likelihood that a question is relevant and that the majority of questions are good for use with minimal expert supervision.

	HumanSum	AutoSum	RawText
Well Formed?	**87.0**	78.0	60.0
Relevant?	84.0	**90.0**	60.0
Correct?	**87.0**	62.0	48.0
Partial Correct?	**98.0**	91.0	95.0

Table 1: Comparison across input method of auto-generated questions (n=100 per method). Numbers represent the %age of the 100 generated questions that the annotators decided had the given attribute. "HumanSum" refers to questions generated from Human-written Summaries, "AutoSum" is from auto-generated summaries, and "RawText" is from the textbook.

	A1	A2	A3	Pairwise Cohen κ
Well Formed?	85.0	83.0	79.0	(0.26, 0.35, 0.46)
Relevant?	86.0	72.0	81.0	(0.30, 0.31, 0.24)
Correct?	92.0	80.0	78.0	(0.35, 0.44, 0.35)
Partial Correct	98.0	88.0	91.0	(0.69, 0.76, 0.73)

Table 2: Comparison across our three annotators (A1, A2, A3) of the evaluation of 100 questions generated from Human-written Summaries. Numbers represent the %age of the 100 generated questions that the given annotator decided had the given attribute. Pairwise Inter-Annotator Agreement is reported in the order (A1-A2, A2-A3, A3-A1).

In Table 2, we report the evaluation results for each annotator and the pairwise Kappa score for inter-annotator agreement. Surprisingly, the inter-annotator agreement is low for the "well-formed" metric which seemed to be the simplest metric. It turns out that the evaluation task is not straightforward. In the next section, we will look at some disagreement examples which are, also, indicative of the errors of the model. It is, however, clear that in future work we should seek to refine the metrics and guidelines we use when asking human annotators to evaluate QG-QA pairs.

4 Disagreements and errors

We will present two types of well-formedness disagreements that were quite common. Q(1) and (2) are two examples of what we called he "wh- in situ" disagreement. The wh-word has not moved

[4]The code we used to run the model (a fork of the original model author's repo with some minor bug fixes and additions) can be found here https://github.com/liamdugan/question_generation

[5]https://huggingface.co/facebook/bart-large-cnn

Figure 3: Blank interface for inputting text and generating questions

to the front to form a proper sentence. One of the two annotators marked this type of questions as non well-formed. Q(3) and Q(3) display a second common source of disagreement. These questions are grammatically correct but pragmatically they can stand on their own because they can only be interpreted with access to the previous context. In our guide for well-formedness we had not anticipated these nuances.

1. Models that assign probability to sequences of a word are called what?
2. What should be tested with many languages, not just one?
3. What lets us match the word if it has the character before the question mark? 4. What indicates the editing operations needed to equalize two strings?

Looking at examples of agreement on the model's failures, we highlight the following two types. Q(1) is representative of a poorly formed question due to poor extraction of special characters. Q(2) is a bad question because of poor understanding of the fact that x (the answer) is not a special type of random variable and Q(3) is not a relevant question because it's not quering an important concept.

1. What type of model is naive Bayes?
2. What random variable is created over all test sets to find if we can rule out H_0?
3. How many main methods of smoothing are studied?

5 System Description

To preserve anonymity, we have removed screenshots that display the name of the tool and we have not shared the live URL. The prototype of our tool is designed to to allow a student or an instructor to enter text (in the large text box) by simply copying and pasting. They can click on the "generate cards" button to initiate the model on the back end which will analyze the text and automatically generate flashcards. Instructors have the option of creating a class and assigning different courses for each class. Each course will contain its own set of flashcards which the instructor can make available to the students or even assign them. The output questions are displayed and the corresponding flashcards are created and they can be flipped to see the answer. The user can interact with the questions to edit them or reject them, as seen Fig. (4). Instructors

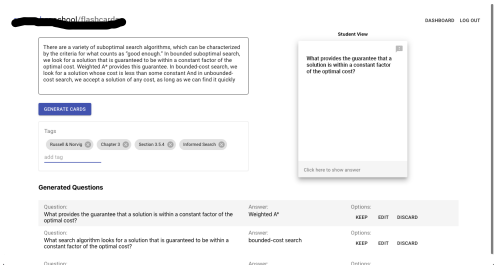

Figure 4: Questions generated from a sample textbook passage

can manage their database of previously generated questions through the following interface. Lastly, here is a sample question that was generated by the model as seen by the students when reviewing that question.

6 Conclusions and Future Work

We presented a tool for generating flash cards automatically to support learning and retention. The evaluation of the models in human and automatically generated studies and the analysis of errors, as revealed in the human evaluations, point to the following next steps: 1) Improve the relevance of questions by fine-tuning the QA model in the education domain. There is a mismatch between SQuAD and the target domain. 2) Improve the quality of questions by generate more high level questions that deepen the understanding of the content. We plan to do with education data ((Ko et al., 2020). 3) Evaluate the quality of the generated questions with questions that have been generated by humans.

References

Manish Agarwal and Prashanth Mannem. 2011. Automatic gap-fill question generation from text books. In *Proceedings of the sixth workshop on innovative use of NLP for building educational applications*, pages 56–64.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. *arXiv preprint arXiv:1906.05416*.

James W Antony, Catarina S Ferreira, Kenneth A Norman, and Maria Wimber. 2017. Retrieval as a fast route to memory consolidation. *Trends in cognitive sciences*, 21(8):573–576.

Arjun Singh Bhatia, Manas Kirti, and Sujan Kumar Saha. 2013. Automatic generation of multiple choice questions using wikipedia. In *International conference on pattern recognition and machine intelligence*, pages 733–738. Springer.

Peter Brown, H Roediger, Mark McDaniel, and Make It Stick. 2014. The science of successful learning. *Cambridge, MA*.

Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. 2018. Learningq: a large-scale dataset for educational question generation. In *Twelfth International AAAI Conference on Web and Social Media*.

Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2019. Reinforcement learning based graph-to-sequence model for natural question generation. *arXiv preprint arXiv:1908.04942*.

Rui Correia, Jorge Baptista, Maxine Eskenazi, and Nuno Mamede. 2012. Automatic generation of cloze question stems. In *International Conference on Computational Processing of the Portuguese Language*, pages 168–178. Springer.

Lorena Antonia Escobar Ibarra and Mirella Del Rosario Wong Martillo. 2017. The use of flashcards for strengthening the reading comprehension. B.S. thesis, Universidad de Guayaquil. Facultad de Filosofía, Letras y Ciencias de la ....

Jonathan M Golding, Nesa E Wasarhaley, and Bradford Fletcher. 2012. The use of flashcards in an introduction to psychology class. *Teaching of Psychology*, 39(3):199–202.

Daniel Jurafsky and James H Martin. 2009. *Speech and language processing (Prentice Hall series in Artificial Intelligence)*. Prentice Hall NJ.

Wei-Jen Ko, Te-Yuan Chen, Yiyan Huang, Greg Durrett, and Junyi Jessy Li. 2020. Inquisitive question generation for high level text comprehension. *arXiv preprint arXiv:2010.01657*.

Ghader Kurdi, Jared Leo, Bijan Parsia, and Salam Al-Emari. 2019. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Mukta Majumder and Sujan Kumar Saha. 2014. Automatic selection of informative sentences: The sentences that can generate multiple choice questions. *Knowledge Management & E-Learning: An International Journal*, 6(4):377–391.

Toshiya Miyatsu, Khuyen Nguyen, and Mark A McDaniel. 2018. Five popular study strategies: Their pitfalls and optimal implementations. *Perspectives on Psychological Science*, 13(3):390–407.

Annamaneni Narendra, Manish Agarwal, and Rakshit Shah. 2013. Automatic cloze-questions generation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 511–515.

Barbara Oakley, Terrence Sejnowski, and Alistair McConville. 2018. *Learning How to Learn*. Penguin.

Sumeet Pannu, Aishwarya Krishna, Shiwani Kumari, Rakesh Patra, and Sujan Kumar Saha. 2018. Automatic generation of fill-in-the-blank questions from history books for school-level evaluation. In *Progress in Computing, Analytics and Networking*, pages 461–469, Singapore. Springer Singapore.

Juan Pino and Maxine Eskenazi. 2009. Measuring hint level in open cloze questions. In *Twenty-Second International FLAIRS Conference*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. Few-shot question answering by pretraining span selection. *arXiv preprint arXiv:2101.00438*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Siyuan Wang, Zhongyu Wei, Zhihao Fan, Zengfeng Huang, Weijian Sun, Qi Zhang, and Xuan-Jing Huang. 2020. Pathqg: Neural question generation from facts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9066–9075.

Kathryn T Wissman, Katherine A Rawson, and Mary A Pyc. 2012. How and when do students use flashcards? *Memory*, 20(6):568–579.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Laura Zavala and Benito Mendoza. 2018. On the use of semantic-based aig to automatically generate programming exercises. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, page 14–19, New York, NY, USA. Association for Computing Machinery.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020. Sg-net: Syntax-guided machine reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9636–9643.