Automatic Multiple Choice Question Generation From Educational Text

By Isaac Tsolak

1. Introduction

Test taking is vitally important to ensure that students are experiencing effective teaching and learning, yet creating high quality multiple choice questions (MCQs) is a time consuming task for educators at all levels. Teachers and professors spend significant time crafting questions that test for comprehension and recall from content learned over the school year in textbooks and other course materials. This time consuming challenge limits the availability of practice materials for students and increases the educator's workload.

Automatic question generation(AQG) addresses this challenge with the use of natural language processing(NLP) and transformer models to generate questions directly from any educational text. The ability to automatically produce MCQs has many applications in a real world educational setting.

- **Personalized Learning**: Generate custom practice questions tailored to specific textbook sections or any other learning objective.
- **Large Scale Assessment**: Enables creation of questions for standardized testing and online courses.
- **Self Study tools**: Can provide students with unlimited practice materials for exam preparation.
- **Content Accessibility**: Helps educators quickly develop tests and quizzes without extensive manual effort.

This project develops an end to end system that takes educational text as input and generates complete multiple choice questions with plausible false responses. The code combines pre-trained question generation with entity extraction and semantic similarity techniques to create understandable answerable questions useful for students and educators for practice problems and time management.

2. Related Work

Automatic question generating is proving to become a useful tool for educators and students alike, this makes it an important research topic for the future in order to make education easier and less time consuming. While professors may be teaching

multiple classes or want to have many versions of the same exam, creating the questions for these tests is very time consuming and they could be focusing on other tasks like research or meeting with students. Automatic question generation can provide the extra time needed to complete more meaningful tasks. Zhang et al. (2021) show a comprehensive survey categorizing generation by input format target answers and question types. This survey provides a framework for understanding the different ways researchers have developed for question generation. In the research done by Zhang et al.(2021) they found that modern approaches are using deep learning methods to create syntactically correct and relevant questions and that it has also evolved from rule based approaches to the use of neural networks for question generation. The research completed from (Soni et al.) found that not only could automatic question generation be beneficial in the educational field but it also has applications to other automated help systems, chat bots and healthcare. In this project the inspiration for creating an interface came from the work being done by (Nwafor and Onyenwe) as they found that the user-friendly interfaces make it simple and accessible for educators or individuals without coding experience. Although this is a helpful application of NLP there are remaining challenges (Zhang et al; 2021; Soni et al.) found that it can be difficult to manage difficulty levels between different individuals and their level of education as well as semantic correctness. This project builds on these established approaches to showcase the ability of these pre-trained models to generate questions for educational purposes, while there is still the need for continued research to improve the question quality and and question difficulty.

3. Methods

This Multiple choice question generation software consists of four main components. The model looks for answer candidates, then generates questions, after the distractor section looks for plausible false answers and finally output formatting.

3.1 Answer Candidate Extraction

The first step in the process of question generation identifies a potential answer that exists with the imputed text. The use of apcCy's en_core_web_sm model for named entity recognition and noun phrase chunking it then processes the text as follows.

1. **Sentence Segmentation**: split input text into individual sentences using spaCy's sentence tokenizer.
2. **Entity and Noun Extraction**: For each sentence it extracts:

    a. Named entities(persons, locations, organizations, etc).
    b. Noun chunks (noun phrases identified by dependency parsing).
3. **Single Word Filtering**: The code then restricts candidates to single word answers to ensure concise answerable questions.
4. **Checks for Duplicates**: Removes duplicate keywords across sentences

This approach looks primarily at factual statements and nouns which are typically suitable answers for educational multiple choice questions. The marking for single word answers reduces complexity in distractor generation while focusing on key terminology.

3.2 Question Generation

For the generation of the questions this project uses the pretrained T5 model (Valhalla/t5-base-qg-hl), which is specifically used for question generation with answer highlighting.

5. **Context Preparation:** Insert <hl> tokens around the answer in the original sentence
6. **Tokenization:** Tokenize the prepared input using T5Tokenizer
7. **Generation:** Use beam search (beam_width=4) with early stopping to generate the question
8. **Decoding:** Decode the output tokens back to natural language text

3.3 Distractor Generation

Distractor generation creates plausible answers that are not correct.

9. **Candidate Pool:** Selects words from the previously extracted keywords in the input passage (not including the correct answer).
10. **Type Matching:** Filters for single word answers to match the answer formatting.
11. **Randomization:** Shuffles distractor candidates for more variety.

While this method is simple it still has answer proximity keywords from the input passage are related and therefore they are reasonable distractors for the correct answer.

3.4 Output and Interface

The final step of this implementation combines the generated questions, correct answers and distractor answers. The system outputs all generated multiple choice questions in an organized structured format suitable for display or exporting. The interface makes the system accessible to educators and students alike without requiring programming knowledge.

4. Data

This project is designed to work with educational text and does not require any training dataset because it uses pre-trained models from hugging face and spaCy. The three models used are the T5 based question generation model valhalla/t5-base-qg-hl with answer highlighting coming from all-MiniLM-L6-v2 sentence embedding model for semantic similarity computation, and spaCy's en_core_web_sim for named entity recognition and noun phrase extraction. All models are publicly available with permissive open-source licenses (Apache 2.0 and MIT).

5. Discussion

6.1 What Worked

The questions generated were of high quality that were grammatically correct and relevant to the educational input passage in most of the outputs. Also the answer highlight method effectively made the model generate questions based on the input passage and targeted the desired information. Overall most questions generated were complicated enough to use as study/practice material for students and exam questions for overworked educators. The interface makes it an easy accessible way for students and educators to generate questions without coding experience which allows for wide range accessibility.

6.2 Limits

The biggest limit of this project is the single word answer constant, this model is not smart enough to generate questions with answers longer than a single word. This issue limits the types of questions that can be generated. Many exams especially at higher education levels require more complex questions with longer explanations for answers leading to simplistic questions for tests. Another limit is the distracting quality, random selection may not be the best way to handle this problem although the implementation is simple some of the generated distractors may be too obvious or too similar to the actual answer leading to misleading answer selection.

7. Conclusion

The goal of this project was to demonstrate the ability of pre-trained NLP models to automatically generate multiple choice questions from educational text. By combining these pre-trained models this project is an end-to -end built system that can assist educators and students in creating practice and real world exam questions for testing and studying. This approach shows alternative ways to reduce busy work for educators and an efficient method for students to study. Although the questions generated could be more informative and complex this project still can create meaningful and answerable questions.

References

Nwafor, C. A., & Onyenwe, I. E. ***An automated multiple-choice question generation using natural language processing techniques***

Rakangor, S., & Ghodasara, Y. R. ***Literature review of automatic question generation systems***

Soni, S., Kumar, P., & Saha, A. ***Automatic question generation: A systematic review***

Zhang, R., Guo, J., Chen, L., Fan, Y., & Cheng, X. (2021). ***A review on question generation from natural language text*** **Association for Computing Machinery (ACM). doi:10.1145/3468889**