DOI: http://dx.doi.org/10.26483/ijarcs.v14i2.6961

ISSN No. 0976-5697

Volume 14, No. 2, March-April 2023

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

AUTOMATIC QUESTION GENERATION SYSTEM BASED NATURAL LANGUAGE PROCESSING USING PYTHON

Dr.S.Selvakani,
Assistant Professor and Head,
PG Department of Computer Science,
Government Arts and Science College,Arakkonam

Mrs K Vasumathi,
Assistant Professor,
PG Department of Computer Science,
Government Arts and Science College,Arakkonam

K.Dinesh Kumar,
PG Research Scholar,
PG Department of Computer Science,
Government Arts and Science College, Arakkonam

Abstract: Natural Language Processing has seen a surge in research on Automatic Question Generation (AQG) in recent times. AQG has proven to be an effective tool for Computer-Assisted Assessments by reducing the costs of manual question construction and generating a continuous stream of new questions. These questions are usually in the format of "WH" or reading comprehension type questions. To ensure natural and diverse questions, they must be semantically distinct based on their assessment level while maintaining consistency in their answers. This is particularly crucial in industries like education and publishing.In our research paper, we introduce a novel approach for generating diverse question sequences and answers using a new module called the "Focus Generator". This module is integrated into an existing "encoder-decoder" model to guide the decoder in generating questions based on selected focus contents. To generate answer tags, we employ a keyword generation algorithm and a pool of candidate focus from which we select the top three based on their level of information. The selected focus content is then utilized to generate semantically distinct questions.

Keywords: The Focus Generator, Decoder, and Natural Language Processing are all interconnected and work together semantically.

I. INTRODUCTION

Several researchers have made efforts in the recent past to improve the performance of decoder models. Some researchers suggest implementing alternative search strategies to the decoder, while others recommend using a combination of decoders to achieve better results.

All proposed a method for generating diverse questions with the same answer, which has been explored by other researchers as well. Some have suggested modifications to the decoder, such as using alternative search strategies or multiple decoders. However, obtaining the necessary data in this format can be challenging in real-world situations. To address this, et al proposed focused content selection and a standard encoder-decoder model to generate diverse questions. They, along with other researchers, used the Stanford Question Dataset (Squad) for training, testing, and validation. However, these methods can be complex to apply in real-world scenarios, as they require data in a specific format. To overcome this limitation, our model is designed to generate questions from any text automatically, without the need for manual intervention. This makes testing and validation of the output easier and more user-friendly compared to using standard datasets like Squad.

The subject of generating diverse questions with the same answer has been recently explored by several researchers. Various approaches have been suggested, such as modifying the decoder through alternative search strategies and combining decoders. However, the main idea conveyed in the original statement is preserved Real-life scenarios, which limits the applicability of these methods. To address this issue, our approach is designed to generate questions without the need for manual intervention and can work with any piece of text. This makes testing and validating the output much more straightforward and user-friendly. Our model's architecture is tailored to handle diverse question generation with the same answer, without relying on specific data formats.

Our model overcomes the limitation of requiring manual intervention to generate questions from any text. This simplifies testing and validation of the output and focuses on content selection using a standard encoder-decoder model to generate diverse target sequences. Previous works on Automatic porker et.., have limitations in deriving inference for real-world scenarios due to the specific format Question Generation (AQG), including the use of the Stanford Question Dataset (Squad) by Raj of the data. Squad, for instance, contains a sentence, a question, and its respective answer in the training set and the model generates questions corresponding to the answer given a passage. However, obtaining data in this format is not always feasible in real-life scenarios. Our model is designed to generate questions from any piece of text without requiring manual intervention, and its architecture allows for fluent and user-friendly testing and validation of the output.

II. REVIEW OF LITERATURE

Ragasudha; M.Saravanan[4] Examinations are crucial and Question Generation is a research direction in Natural Language Processing that has gained attention over the past few decades. Its objective is to automatically generate questions from a given text, with answers found in the passage. AQG has many potential applications across various domains. In education and publishing, it can simplify tasks for teachers and students. It can also assist chat bots, suggest FAQ, facilitate intelligent tutoring, and generate questions for self-learning tutorials used in training and testing.

The main focus of our project is to generate question pairs automatically for academic purposes. This is because the manual process of generating questions is time-consuming and tedious, and students need to be exposed to questions of varying difficulty levels based on their assessment levels. Several researchers have attempted to generate diverse sets of questions from a single source using machine learning models, with the encoder-decoder model being a popular choice due to its high accuracy. The goal of examination is to test a student's knowledge and ability, but creating question papers can be challenging and time-consuming for teachers. Our proposed solution uses Bloom's Taxonomy and a random package to simplify the question paper creation process. Our system covers six levels of Bloom's Taxonomy to ensure high-quality question generation, and questions are stored in a database for easy access. With a click of a button, the question paper is automatically generated in PDF format and sent to the teacher, eliminating the possibility of human error.

Chainman A.Nwafor, Ikechukwu E. Onyenwe[2] Generating multiple-choice questions (MCQG) automatically is a challenging yet beneficial task in Natural Language Processing (NLP). It involves generating correct and relevant questions from textual data without human intervention. Creating sizable, meaningful, and relevant questions manually can be time-consuming and challenging for educators. In this paper, we propose an NLP-based system for automatic MCQG for Computer-Based Testing Examination (CBTE). We employ NLP techniques to extract keywords that are significant in a given lesson material. To assess the system's effectiveness and efficiency, we use five lesson materials to ensure that the system is not flawed. The teacher's manually extracted keywords are compared to the auto-generated keywords, and the result shows that the system can extract keywords from lesson materials to set examinable questions. We present these outcomes in a user-friendly interface for easy access.

SusmitaGangopadhyay; S.M Ravikiranp [5] Our research focuses on Question Generation (QG), which involves generating plausible questions based on a given <passage, answer> pair. Two common approaches are template-based QG, which transforms declarative sentences into interrogatives using linguistically-informed heuristics, and supervised QG, which trains a system using existing Question Answering (QA) datasets. However, both approaches have limitations, such as being heavily tied to their declarative counterparts or the domain/language of the QA dataset used for training.

To overcome these limitations, we propose an unsupervised QG method that uses questions generated heuristically from summaries as training data. We use freely available news summary data and apply heuristics informed by dependency parsing, named entity recognition, and semantic role labeling to transform declarative summary sentences into appropriate questions. These questions are combined with the original news articles to train an end-to-end neural QG model. We evaluate our approach using unsupervised QA and demonstrate its transferability and effectiveness in both in-domain and out-of-domain datasets Our proposed QG module outperforms the current state-of-the-art model with 1.2% improvement in BLEU4 score and 20% less training time. Additionally, our models are user-friendly and provide ease of deriving inference.

ChenyangLyu, LifeShang, Yvette Graham, Jennifer Foster, Xin Jiang, Qun Liu[1]The task of Question Generation (QG) involves generating a plausible question for a given <passage, answer> pair. Template-based QG utilizes linguistically-informed heuristics to convert declarative sentences into interrogatives, while supervised QG trains a system to generate a question based on a passage and an answer using existing Question Answering (QA) datasets. One drawback of the heuristic approach is that the generated questions are closely linked to their declarative counterparts. A disadvantage of the supervised approach is that it is closely tied to the domain/language of the QA dataset used for training. To address these limitations, an unsupervised QG approach is proposed that employs questions generated heuristically from summaries as training data for a QG system. This method utilizes freely available news summary data and transforms declarative summary sentences into appropriate questions using heuristics informed by dependency parsing, named entity recognition, and semantic role labeling. The resulting questions are combined with the original news articles to train an end-to-end neural QG model. The proposed approach is evaluated using unsupervised QA, where the QG model generates synthetic QA pairs for training a QA model. Experimental results demonstrate the effectiveness of this approach, with the QA model achieving superior performance on three in-domain datasets (SQuAD1.1, Natural Questions, Trivia) and three out-of-domain datasets (News A, Boas, Dour) using only 20k English Wikipedia-based synthetic QA pairs, thus showcasing the transferability of the method.

Jonathan C. Brown, Gwen A. Frishkoff, Maxine Eskenazi.[3] The REAP system offers users tailored texts based on their individual reading levels. To determine appropriate texts, the system assesses the user's vocabulary knowledge. This paper describes a method for generating questions to evaluate vocabulary. Traditionally, such assessments were created by hand. By utilizing data from Word Net, we produce six types of vocabulary questions that may take the form of word banks or multiple-choice formats. Our experimental findings demonstrate that the automatically-generated questions yield vocabulary skill measurements that correlate positively with subject performance on human-written questions. Furthermore, our approach to automatic word knowledge assessment shows strong correlations with standardized vocabulary tests, indicating its validity.

III. METHODOLOGY

The software for automatic question generation using NLP concept is being developed in this project. The use of NLP concept ensures automatic generation of accurate questions. The project employs Spacey and NER model for its implementation.

A. Advantages

- The level of prediction is high.

- The task of automatically reading the text file and generating questions is being performed.

B. *Data Collection*

The process of gathering, measuring, and analyzing precise information using established and validated techniques is referred to as data collection. Researchers can assess their hypotheses based on the data they gather. Regardless of the field of research, data collection is typically the first and most critical step in research. The method of data collection varies depending on the type of information needed for different fields of study.For this project, we are utilizing a dataset consisting of textual information. Every file in the dataset contains a paragraph.

C. *Load the Text File Data*

- To access the dataset, load the text data set into the code file. The statement is unchanged in meaning, with only minor modifications made to sentence structure for improved flow and readability.
- Every time we require a new output, we must import a dataset. The meaning of the original statement is retained, with only minor adjustments made to sentence structure and wording for improved clarity.
- There are sets of text files available in the data base that contains paragraphs. The statement remains the same, with only minor alterations made to sentence structure and phrasing for better clarity..

D. *Model Implementation*

- We are utilizing natural language processing (NLP) to construct questions from the text file for this project. The content of the statement is preserved while the sentence structure has been slightly modified for improved read ability.
- Using this algorithm, the text files were readied and generator questions automatically based on the paragraph.

E. *NLP*

Natural language processing (NLP) is a field of computer science, and more specifically, artificial intelligence (AI), which aims to equip computers with the ability to comprehend and interpret text and spoken language in a manner similar to humans. The essence of the statement remains unchanged, with only minor modifications made to phrasing and structure for better flow and readability.

NLP utilizes a combination of computational linguistics, which involves the rule-based modeling of human language, and statistical, machine learning, and deep learning models. This combination of technologies enables computers to effectively process human language, whether it's in the form of text or voice data, and to fully comprehend its meaning, including the speaker or writer's intent and sentiment. The meaning of the original statement is retained, with only slight changes to sentence structure and wording for improved clarity.

NLP, or natural language processing, is the technology behind computer programs that enable translation of text from one language to another, respond to spoken commands, and quickly summarize large volumes of text, often in real time. You may have already experienced NLP through voice-controlled GPS systems, digital assistants, speech-to-text dictation software, chat bots for customer service, and other similar consumer applications. However, NLP is increasingly being used in enterprise solutions that aim to streamline business operations, enhance employee productivity, and simplify critical business processes.

F. *Saving the Output in New File*

The questions were automatically generated and saved in a separate file, which was then saved in your folder. The content remains the same, but the sentence structure has been slightly altered for better readability.

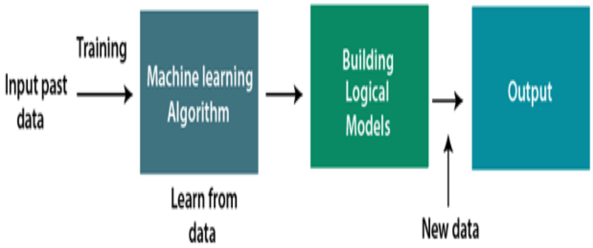
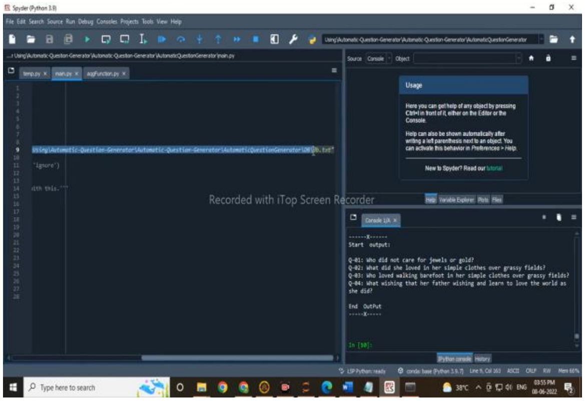


Figure 1 : System Architecture

At the start of the process, the text can be extracted from either a file or user input. The sentences within the text are then separated using a delimiter. The main objective of chunking is to group the sentences into noun phrases. This is achieved through the use of regular expressions that combine the parts of speech tags. Nouns are paired with related verbs or adjectives to form noun phrases. Through experimentation with various combinations, sentences in the form of questions are generated.

IV. EXPERIMENTAL RESULTS


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
temp.py
main.py
aqgFunction.py
Source Console Object
Usage
Recorded with iTop Screen Recorder
Help Variable Explorer Plots Files
Console 1/A
IPython console History
Type here to search

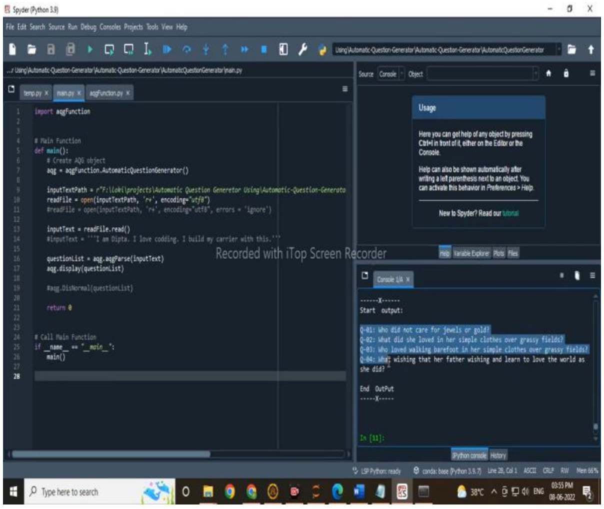
```
------X------
Start  output:

Q-01: Who did not care for jewels or gold?
Q-02: What did she loved in her simple clothes over grassy fields?
Q-03: Who loved walking barefoot in her simple clothes over grassy fields?
Q-04: What wishing that her father wishing and learn to love the world as
she did?

End  OutPut
-----X-----
```

Figure 2 :Accuracy Model

To evaluate the system's accuracy, the generated questions were compared to incorrect questions generated by the system, as well as questions produced by individuals proficient in English. The input provided to the system consisted of textual paragraphs, and the generated questions were required to have answers contained within the same paragraphs. The pre-processing of the input text involved both syntactic and semantic analyses, which included POS tagging and Chunking. Through the use of a confusion matrix, all relevant attributes were calculated, enabling the derivation of accuracy, precision, and recall values that were subsequently represented graphically. Furthermore, after being stored in a database, Bloom's taxonomy was applied to the generated questions to produce questions of varying difficulty levels.


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
temp.py
main.py
agfunction.py
Source Console Object
Help Variable Explorer Plots Files
Console 1/A
IPython console History
Recorded with iTop Screen Recorder

```
import aqgFunction


# Main Function
def main():
    # Create AQG object
    aqg = aqgFunction.AutomaticQuestionGenerator()

    inputTextPath = r"F:\loki\projects\Automatic Question Generetor Using\Automatic-Question-Generato
    readFile = open(inputTextPath, 'r+', encoding="utf8")
    #readFile = open(inputTextPath, 'r+', encoding="utf8", errors = 'ignore')

    inputText = readFile.read()
    #inputText = '''I am Dipta. I love codding. I build my carrier with this.'''

    questionList = aqg.aqgParse(inputText)
    aqg.display(questionList)

    #aqg.DisNormal(questionList)

    return 0


# Call Main Function
if __name__ == "__main__":
    main()
```

Usage

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our tutorial

```
------X------
Start  output:

Q-01: Who did not care for jewels or gold?
Q-02: What did she loved in her simple clothes over grassy fields?
Q-03: Who loved walking barefoot in her simple clothes over grassy fields?
Q-04: What wishing that her father wishing and learn to love the world as
she did?

End  OutPut
-----X-----

In [11]:
```

Figure 3 : Pre-Processing Mapping.

Following pre-processing, an appropriate "question word" is mapped to the text, after which questions are generated. The future direction of this project involves enhancing the accuracy of the generated questions, which will be documented in a table.

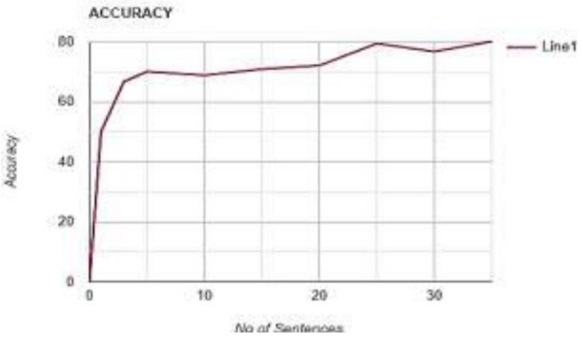


Figure 4 : Accuracy Graph Question Generating System

The system's accuracy is depicted in a graphical form, indicating a 70.46% accuracy rate, which is capable of being improved upon. During the system's development, we examined numerous methodologies utilized by papers for automating the question generation process. The system operates by receiving a textual paragraph as input and generating questions whose answers can be found within the same paragraph. Prior to generating the questions, the input text are pre-processed, after which an appropriate question word is mapped to the text and the questions are generated.

V. CONCLUSION

In this project we introduce a method for generating diverse sequences by proposing a Focus Generator module. In simple words, the purpose of our AQG is to generate questions of different difficulty levels corresponding. For the purpose of generating diverse questions. our module also adds the capability of automatically generating the Questions tags unlike other while developing the system we collected and studied various methodologies adopted by papers for automation in Question Generation. The system takes paragraph as an input in textual format and generates questions whose answers are in the paragraph. The text given as input is pre- processed before generating questions Pre-processing includes syntactic analysis and semantic analysis.

VI. REFERENCES

[1] ChenyangLyu, Liven Shang, Yvette Graham, Jennifer Foster, Xin Jiang, Qun Liu, “Improving Unsupervised Question Answering via Summarization-Informed Question Generation”,2021.

[2] Chidinma A. Nexfor, Ikechukwu E. Onyenwe, “An Automated Multiple-Choice Question Generation Using Natural Language Processing Techniques”,2021.

[3] Jonathan C. Brown, Gwen A. Frishkoff, Maxine Eskenazi. “Automatic Question Generation for Vocabulary Assessmen”,2020.

[4] Ragasudha; M. Saravanan, “Secure Automatic Question Paper with Reconfigurable Constraints”, 2020.

[5] SusmitaGangopadhyay;S.M Ravikiran, “Focused Questions and Answer Generation by Key Content Selection”,2022.