Journal of Advanced Zoology

ISSN: 0253-7214
*Volume **44** Issue **S-6 Year 2023** Page **747:756***

Automated Questions Unique Arrangement (A.Q.U.A)

Akanksha Sharma[1], Saket Savarn[2], Adarsh Anand[3], Sunil Kumar S. Manvi[4]

[1,2,3,4]Department of CSE, Reva University.
Email: akankshasharma1316@gmail.com[1], saketsavarn07@gmail.com[2], adarshanshu7@gmail.com[3], ssmanvi@reva.edu.in[4]

**Corresponding author's E-mail: akankshasharma1316@gmail.com*

Article History

Received: 06 June 2023
Revised: 05 Sept 2023
Accepted: 30 Nov 2023

Abstract

With the world digitizing and moving at a fast pace, framing questions for examinations or learning is a time-consuming process and requires a lot of critical thinking. Questions we solve in the exams, for instance, school and college level examinations, are similar to the last year papers and contain repeated questions with little or no paraphrasing or modifications. Educators spend a significant amount of time in preparing question papers to come up with creative brainstorming questions. Automation has become a vital aspect of life. New technologies are coming up every day to minimize manual work and make everything automated with just a click. Considering the present pandemic scenario, education is now internet based and exams are being conducted online. Most of the examinations are based on multiple choice questions and these questions are, in most cases, taken from popular quizzing websites. This practice makes it easier for students to find the correct answer without even studying the subject and increases malpractices. We propose an automatic solution to the issue of making questions that will save time and energy and also promote proper learning with our model "A.Q.U.A – Automated Questions Unique Arrangement. It is a machine learning model that uses transformers for natural language processing and generating meaningful and understandable questions from the given context. A.Q.U.A will be of great use in online assessments , school level and university level exams, as well as competitive exams. It'll be also helpful for students and learners to take practise tests for a topic and evaluate their knowledge in it.

Keywords: *Question Generator, Machine Learning, Transformers, Natural Language Processing*

1. Introduction

In order to learn a subject properly, the knowledge gained needs to be assessed by answering questions ourselves by answering questions. Examinations are conducted for the same purpose. E-learning MOOCs (Massive Open Online Courses) had gained popularity in the past few years but with the onset of pandemic even the traditional school and college culture shifted to online platforms. Multiple choice questions are now preferred over descriptive questions. But making questions for examinations, no matter the subject or degree, is always time-taking and stressful. Teachers spend a significant amount of time and energy in doing so and have to ensure that important topics are not missed. This leads to them choosing questions from popular quizzing websites or from previous years papers. Virtual examinations have thus increased the risk of malpractices as it's easy for students to find out solutions from the internet. Also, if seen from a technology point of view, question generation is a manual process and hence can be erroneous and lengthy. To deal with these disadvantages of manual question generation and to lessen the burden for teachers, there is a need for an automatic question generator that generates meaningful questions which can be used to test the students. Therefore, in this paper we have proposed our model A.Q.U.A – Automated Questions Unique Arrangement. A.Q.U.A. creates questions from a given context or study material that can be a text file, pdf file, word file, jpg/jpeg file or any other format file. The input text is analyzed, and processes and questions are randomly generated from the sentences. Our model creates Multiple Choice Questions and Boolean type Questions from given context and also paraphrases the given question.

Literature Survey

Our first understanding of natural question generation comes from breaking all the data inputted in sentences and paragraphs and then to maximize the no. of questions that can be generated from that one sentence or paragraph in the most natural way while making sure that generated question isn't just syntactic transformation and uses synonym [1], hence from [1] it is clear that just syntactic transformation of declarative sentences to interrogative format is not enough for question generation. As we know, text generation of any kind is usually done by creating tokens or tags of each word which is then treated as graph data structures and then using Breadth or depth-first search, we form combinations until a grammatically correct sentence is formed. For this purpose, Beam-Search Algorithm has been used a lot, but as per [2] the issues faced by BS algorithm is that the generated output isn't diversified and similar in context, which can be an issue in question generation as well if the questions generated aren't unique. So, the authors in [2] came up with "Diversified Beam Search" which makes generating text of diverse nature with more probability. Another thing to point out is that questions often revolve around Nouns and to spot them using standard methods can fail often as Nouns can be very varied in nature as per the context like Names, Places, etc. Hence, using NER method (Named Entity Recognition) as shown in [3] with KWS method (Keyword Spotting System) as shown in 4 can be really helpful in simplifying the process, even though [3] & [4] where used for Google Assistants, on changing the corpus and making changes to the algorithm, the same should be working for Question Generation as well and the success or failure of it is one of the agendas we are looking forward to.

One great possibility of generating questions is greatly shown in [5] where they follow multiple steps such as "sentence simplification" which included simplifying complex sentences, then generating "answer phrases" which means to generate answers first rather than questions, then "decomposition of the main verb" followed by "reframing the sentence" and finally using "Statistical ranking" to make the most acceptable questions. This is greatly explained in fig. 1 in [5] and shows great scope in our project as well. The proposed system in [6] overcomes the problem of implicit definitions by using Supervised Learning Approach, Naïve Bayes method. It consists of: Stemmer, Key phrases Extractor, Phrase Mapper, Summarization, Noun Filtering and Question Generator. The set back in this paper is that using phrase mapper needs the user to give the document contents which causes overhead to the system.

Template matching technique focuses on structured data where the different tuples are assigned under column names and then questions can be formed based on the column names. Dynamic tagger is used for classifying the tuples in [7]. The questions which are generated should be ranked according to semantic correctness. The author as described in [8] has proposed this model to generate fill in the blanks, mcqs, true and false type questions. The author encounters paraphrasing and lexical errors in this model. [9] proposes a model for generating questions for nonverbal reasoning type exams. Every question can have similar looking but just one unique solution. The similar options provided are called distractors and the author has described their generation as well. The paper [10] proposes a model called Questionator that uses Convolutional Neural Network and Long Short Term Memory (LSTM). . It mainly comprises three sections namely Image Captioning, Question Generation and the Option Generation. Paper [11] proposes Sentence-to-Question Generation in which the model is divided into three modules ,first, Data Preprocessing, in which the TREC Dataset is parsed and passed to the Named Entity (NE) tagger

and the Parts of Speech (POS) tagger. Second, Elementary Sentence Construction in which the noun phrases, verb phrases and prepositions are used to form elementary sentences. Third, Sentence Classification and Question Generation is done using NE and POS Tag and classified into fine classes and for each class the respective interrogative pronoun is used. In [12], the paper proposes Diverse Beam as an alternative to Beam Search Algorithm, since it requires less memory overhead and computation. Beam search lacks diversity whereas Diverse Beam Search has diversity between beams at each step. DBS is a modification of BS with doubly greedy approximations. The proposed method was analyzed by generating labels from images and it was observed that as the complexity of images increased, DBS performed better than standard beam search. The paper [13] proposes a framework ERNIE-GEN which uses a pre-trained language generation model. It includes sentence generation mechanisms and noise-aware generation methods. It also uses a span generation task to generate text like human writing. Paper[14] establishes that Transformers are better for sequence models than Recurrent Neural Networks(RNN) like Long Short Term Memory(LSTM) . Transformers rely on self-attention rather than the sequence and are faster with better performance. The architecture of the transformer : encoder, decoder and attention stacks are explained in the paper stating their significance.

In paper[15] the authors have created the BoolQ dataset which is one of the best datasets to train a model for generating questions of yes/no type . They have used transfer learning in which they gained more accuracy by training their model first on a larger dataset and then fine tuning on their BoolQ data.

Proposed Work

Our proposed model A.Q.U.A is divided into four parts , first converting input file to text file, second generating multiple choice questions, third , generating boolean or yes/no type questions and fourth, paraphrasing the given question . We are able to achieve these using Natural Language Processing. Our first step involves us to process read-only and image related documents like PDFs mostly, to deal with these types of files we are using Open Computer Vision (OpenCV) and Image Processing algorithms to retrieve text from the PDF files to start processing our texts [16-21]. Before processing our texts, we clean them to remove irrelevant information like page numbers, etc and large heading texts, and we tag our data into tagged data set like nouns, pronouns, verbs, etc. After this we move on to the next step which involves transforming & paraphrasing a normal sentence to question format.

Unlike other similar projects, our project uses Transformers like T5, a text - to - text framework that is based on BERT (Bidirectional Encoder Representations from Transformers) as it's core. Since BERT is a bidirectional encoder, it is efficient at paraphrasing more complex versions as it is encoding from both left & right side of the given input. Over this complex paraphrasing, T5 is an unified text-to-text transformer where both the input & output is text and not a class or encoded/tagged data. This unification allows to use the same model for QnA, Summarization, translation, paraphrasing, predicting next sentence, text generation, etc. Once we have successfully started generating the questions, we sort the questions based on the difficulty level they offer, from Easy, Medium to Hard and divide the question in whatever ratio the user desires for each of these levels of questions like 3:5:2

For 3 Easy, 5 medium & 2 hard questions. The difficulty is decided by sending the question to Google search in backend and seeing how many google search match with the question, the more searches that match, the easier this question is to solve & find hence it's mark as easier. Once the questions have been sorted the user can download the MCQ paper. The user can also decide to get Yes/No Boolean based MCQ questions if needed. We plan to train for multiple epochs with a reward system based on feedback from the user, if the user likes the questions generated then the model is rewarded but if the user has suggestions he/she can choose from a dropdown the issues they faced and the model can work on those parameters in the next epoch but obviously since the question generated is different for different resource file, this may or may not be effective method. We plan to compare two models, one without feedback system and one with feedback system to see which works better for the same resource file. We have hosted this whole project as a web app using the flask python web frame work and offer as a service for teachers & university professors.

Fig 1: Flowchart of the Proposed Work

Implementation

Three different databases are used to train our model .To train A.Q.U.A to form Multiple Choice Questions from a given passage, we've used the SQuAD 1.0 Dataset. SQuAD or Stanford Question Answering Dataset 1.0 contains about 100,000 questions posed by crowdworkers on Wikipedia articles.

For training A.Q.U.A to generate the Boolean Questions, we have used the BoolQ dataset, containing 15942 examples each consisting of three .jsonl files, where each line is a JSON dictionary having question, passage , answer and title. The format of the dictionary looks like as below:

{ "question": "is france the same timezone as the uk", "passage": "At the Liberation of France in the summer of 1944, Metropolitan France kept GMT+2 as it was the time then used by the Allies (British Double Summer Time). In the winter of 1944--1945, Metropolitan France switched to GMT+1, same as in the United Kingdom, and switched again to GMT+2 in April 1945 like its British ally. In September 1945, Metropolitan France returned to GMT+1 (pre-war summer time), which the British had already done in July 1945. Metropolitan France was officially scheduled to return to GMT+0 on November 18, 1945 (the British returned to GMT+0 in on October 7, 1945), but the French government canceled the decision on November 5, 1945, and GMT+1 has since then remained the official time of Metropolitan France."

"answer": false, "title": "Time in France",}

The other database that we've used to train the paraphrasing part is Quora's Duplicate Question Pairs which has the data in the following form:

id	qid1	qid2	question1	question2	is_duplicate
447	895	896	What are natural numbers?	What is a least natural number?	0
1518	3037	3038	Which pizzas are the most popularly ordered pizzas on Domino's menu?	How many calories does a Dominos pizza have?	0
3272	6542	6543	How do you start a bakery?	How can one start a bakery business?	1
3362	6722	6723	Should I learn python or Java first?	If I had to choose between learning Java and Python, what should I choose to learn first?	1

Fig 2: Snapshot of the Quora Duplicate Question Pairs Dataset

Besides this the T5 transformer used is trained on C4 Corpus which is a big dataset. The following figure represents the components of the implementation of the model which are described in the following paragraphs.

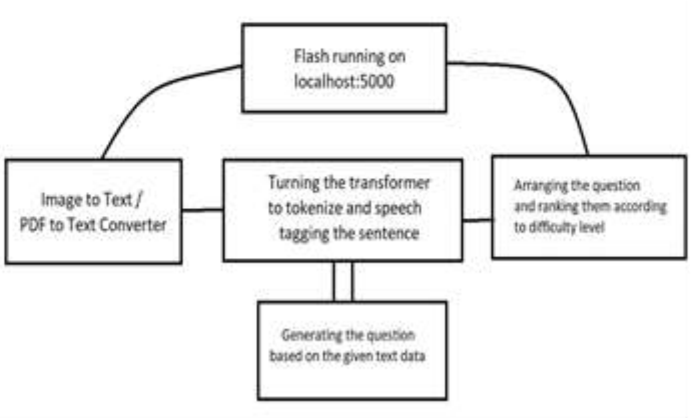

Fig 3 : System Architecture Diagram

A. *Natural Language Processing*

NLP or Natural Language Processing is a branch of Artificial Intelligence that focuses on making a computer understand the human languages.

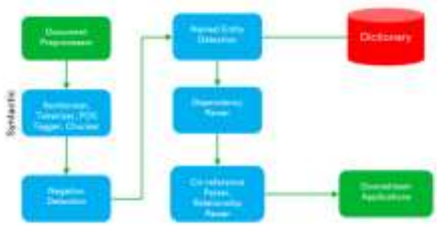

Fig 4: Natural Language Processing components

The main objective of NLP is making the system capable of understanding the meaning of sentences or identifying the underlying pattern between similar sentences in order to carry out the required action by lemmatizing, part-of-speech tagging, stemming and reducing complex sentences into simpler ones. NLP is used for language translation, speech recognition, chatbots etc.

B. *Converting input pdf/image file to text*

The study material or context to generate questions form may or may not be in text format. If the input file is an image file , we first need to convert the image file to text file. This can be achieved using Pytesseract or Python-tesseract . Pytesseract is a python tool for Optical Character Recognition (OCR). OCR involves scanning and analysing an image to detect or identify text encodings in it. The image is converted to grayscale, the text is localized and then the characters are identified. If the input file is a .pdf file , we convert it to image and then extract the text.

Fig 5: OCR input and output snapshots

C. *Generating Multiple Choice Questions*

A.Q.U.A is capable of generating MCQs with distractors i.e. wrong options from a given context.We have taken paragraphs from the SQuAD dataset as input context for our model. For this we require the gensim library, which is a python open source library topic modelling, document indexing and similarity retrieval. Gensim is capable of creating a dictionary using the "corpora.Dictionary( )" function where each sentence split from the input document is provided as a parameter. spaCy is another library used for advanced NLP and has better performance than NLTK . The job of summarizing the context is done by BERT Extractive Summarizer and we can achieve the summary by tuning the ratio, min length and max length parameters. PKE - Python Keyword Extractor is used to extract such words from the context that are also present in the summary around which we can base our question. Nouns are a good fit for such words. Sentence mapping is then done to identify the sentences associated with the generated keywords. Now to generate distractors, which are the wrong options for our MCQ, we use Wordnet. Wordnet is an English lexical database that helps identify the established relationship between words and also gets the correct sense of a word in a sentence. The concept of hypernyms and hyponyms is used to generate these distractors. Combining all these steps , we are able to generate MCQs.

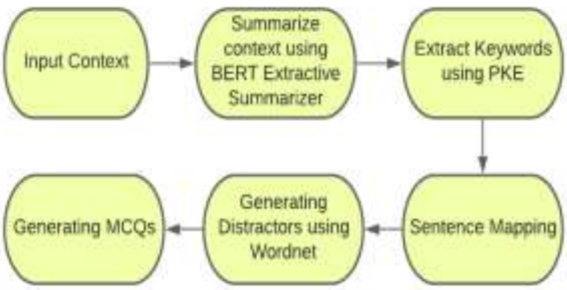

Fig 6: Flowchart for Generating MCQs


```
Running model for generation
 Sense2vec_distractors successful for word :  cricketer
 Sense2vec_distractors successful for word :  india
 Sense2vec_distractors successful for word :  batsmen
{'questions': [{'answer': 'cricketer',
                'context': 'Sachin Ramesh Tendulkar is a former international '
                           'cricketer from India and a former captain of the '
                           'Indian national team.',
                'extra_options': ['Mark Waugh',
                                  'Sharma',
                                  'Ricky Ponting',
                                  'Afridi',
                                  'Kohli',
                                  'Dhoni'],
                'id': 1,
                'options': ['Brett Lee', 'Footballer', 'International Cricket'],
                'options_algorithm': 'sense2vec',
                'question_statement': "What is Sachin Ramesh Tendulkar's "
                                      'career?',
                'question_type': 'MCQ'},
               {'answer': 'india',
                'context': 'Sachin Ramesh Tendulkar is a former international '
                           'cricketer from India and a former captain of the '
                           'Indian national team.',
                'extra_options': ['Pakistan',
                                  'South Korea',
                                  'Nepal',
                                  'Philippines',
                                  'Zimbabwe'],
                'id': 2,
                'options': ['Bangladesh', 'Indonesia', 'China'],
                'options_algorithm': 'sense2vec',
                'question_statement': 'Where is Sachin Ramesh Tendulkar from?',
                'question_type': 'MCQ'},
               {'answer': 'batsmen',
                'context': 'He is widely regarded as one of the greatest '
                           'batsmen in the history of cricket.',
                'extra_options': ['Ashwin', 'Dhoni', 'Afridi', 'Death Overs'],
                'id': 3,
                'options': ['Bowlers', 'Wickets', 'Mccullum'],
                'options_algorithm': 'sense2vec',
                'question_statement': 'What is the best cricketer?',
                'question_type': 'MCQ'}],
 'statement': 'Sachin Ramesh Tendulkar is a former international cricketer '
              'from India and a former captain of the Indian national team. He '
              'is widely regarded as one of the greatest batsmen in the '
              'history of cricket. He is the highest run scorer of all time in '
              'International cricket.',
```


Fig 7 : Generated Multiple Choice Questions

D. *Generating Boolean Questions*

Another kind of popular question format is Boolean or Yes/No type of questions . To achieve this we use the T5 Text-to-Text Transformers model from Huggingface's Transformers library and as mentioned above the database we have used for training it is the BoolQ dataset built by researchers at Google. T5 Text-to-Text Transformer : T5 Transformer proposes a text - to - text framework in which every task that's performed , whether it be language translation, answering questions etc, the input is given in text format and the model is trained to generate output which is also in text format .T5 model helps in achieving state-of-art results in Natural Language Processing. Transformer basically consists of an encoder, decoder and attention block. The input embedding , embedding space in which similar words are grouped together, and positional encoding, consisting of vectors that give context based on the position of the word are passed to the encoder. The attention block calculates how relevant a word in a sentence is to other words in the same sentence and computes the attention vector. Feed Forward Network is then used to transform attention vector in a form that can be accepted by the encoder/decoder block. The output vector and positional encoding is then passed to decoder which adds a masked input that performs attention over output of the encoding stack. We also use the pytorch library which is an open source python library for natural language processing developed by Facebook's AI Research lab. For training the model, we fine tune the T5 model and pass the questions and passage from boolQ dataset as input and make our model learn by stating multiple epochs. On testing, our model was able to generate boolean questions from any given context.

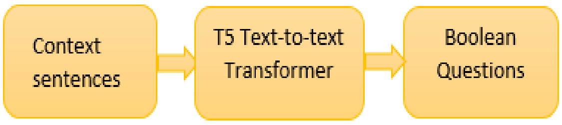

Fig 8: Flowchart for Generating Boolean Questions

```
input_context = {
        "input_text": "Plants have the unique ability of producing their own food through a process called photosynthesis."
    }

output = qe.predict_boolq(input_context)
pprint (output)

{'Boolean Questions': ['Can plants produce their own food?',
                       'Is it true that plants produce their own food?',
                       'Do plants have the ability to produce their own food?'],
 'Count': 4,
 'Text': 'Plants have the unique ability of producing their own food through a '
         'process called photosynthesis.'}
```

Fig 9: Generated Boolean Questions

E. *Paraphrasing Input Questions*

Paraphrasing or modifying questions in such a way that the usage of words change but the meaning remains the same is also a necessary requirement that a question generator should fulfill. The use case of paraphrasing can be when teachers refer to old question papers to get questions on a topic but don’t want to use the exact same question. Rephrasing the question manually can be time consuming so we can use A.Q.U.A to do so. We need to give our model the original question and it will give us a variety of options by paraphrasing it.

The dataset that we have used for this purpose is Quora’s Duplicate Question Pairs dataset and chosen only those pairs that are duplicate ,i.e, have the is_duplicate values as 1 . This data is passed as input to our T5 Text-to -Text transformer model and the model is made to learn by iterating through multiple epochs. The transformer understands that to perform paraphrasing , it needs to use similar words or synonyms that maintain the meaning of the question. The model is now able to paraphrase any given question. For each input question, the model generates as many alternatives as specified by the user by setting the value for the max_questions variable.

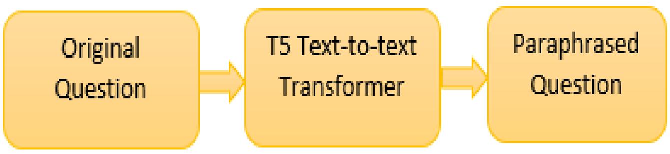

Fig 10: Flowchart for Paraphrasing Questions

```
input_ques = {
    "input_text" : "What is Sachin Tendulkar profession?",
    "max_questions": 5
}
```

```
output = qg.paraphrase(input_ques)
print ("Output :")
print (output)
```

```
0: ParaphrasedTarget: What is Sachin Tendulkar's profession?
1: ParaphrasedTarget: What is Sachin Tendulkar's career?
2: ParaphrasedTarget: What is Sachin Tendulkar's job?
3: ParaphrasedTarget: What is Sachin Tendulkar?
4: ParaphrasedTarget: What is Sachin Tendulkar's occupation?
Output :
{'Question': 'What is Sachin Tendulkar profession?', 'Count': 5, 'Paraphrased Questions': ["ParaphrasedTarget: What is Sachin Tendul
```

Fig 11: Paraphrased Questions

This is the demonstration of our web app using flask web framework, where we are able to upload a resource file (PDF) and convert it into text in the backend, generate the questions and send MCQs. Flask is run on localhost by installing the software XAMPP and activating the apache server, which triggers the localhost to run on ip: localhost: 5000 in the browser. Once the PDF file is uploaded by the user, the python scripts run in the backend that convert the pdf to text, then on the user's choice generate the boolean or mcq questions which then are generated. Since the backend is running on flask as it's easy to automate with python .The output which are the generated questions are put into a text file which it's then available for the user to download [16-21].

Fig 12: Demo of Web App for downloading MCQs.

F. *Modules/Tools Used*

Image Processing:

1. OpenCV [OpenComputerVision for Image Process.]
2. Camelot [detecting tabular data]
3. pyPDF [reading PDF files]
4. PIL [for image processing & manipulation]
5. pdf2image [converting pdf pages to image]
6. pytesseract [extracting text from image]

Data Manipulation & display:

1. Numpy [storing pixels in numpy matrices]
2. Pandas [data manipulation & handling]
3. Xlsxwriter [converting excel sheets]
4. matplotlib [data visualization]

Transformers & nlp:

1. GPT2 [Transformer NN]

2. torch & torch.util [NN]

3. nltk [for Natural Language Processing]

4. t5 [Transformer NN for QnA generation]

3. Results and Discussion

Our model A.Q.U.A is able to generate Multiple Choice Questions, Boolean Type Questions and also paraphrase the questions. The questions generated are meaningful and grammatically correct to a large extent. We observed that MCQs can be generated from a larger context but for generating Boolean Questions, if we give multiple sentences, it considers only one sentence and generates a yes/no type question from that sentence rather than generating one from every sentence. The distractors or wrong options generated by the model for the mcqs are quite valid as they are similar to the answer. With respect to the paraphrasing of the question, the model is able to understand what words of the question seem important and can be replaced with synonyms or other words that will not change the overall meaning of the context. The user can specify a number of questions and A.Q.U.A generates those many alternatives for the user. If we consider the training time. We ran our model on different machines with different configurations on cloud platform and observed the time efficiency. The below graph is plotted based on the model's performance on those machines.

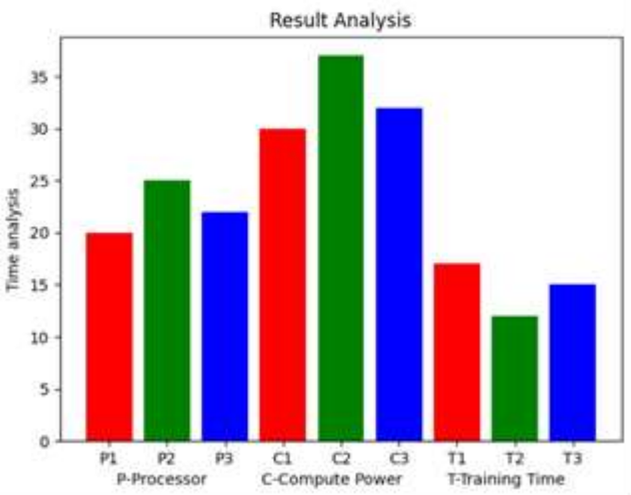

Fig 13: Result Analysis Graph

4. Conclusion

The key challenges in this project were forming meaningful questions that are grammatically correct and making the model understand which kind of key words it needs to focus on. Questions should be sensible and diverse. Question Generation can be used in many scenarios, such as automatic tutoring systems (such as a chatbot generating questions based on a topic being taught), automated Question Paper generation and automated online quizzes. It will be really useful in fields of education for schools and colleges and competitive exams and interviews and MOOC websites. A.Q.U.A can also be used for self learning and practise by students.

Future Scope

The proposed model for the question generation A.Q.U.A can be modified further for generating fill in the blank's kind of questions and also create questions relevant to other subjects like mathematics that involves numbers and symbols and chemistry that includes chemical formula with subscript and superscript. We can also rank questions into Easy, Medium and Difficult by finding its popularity using web scraping. We can also rank the questions specific to the learner to create personalized question sequencing for online courses. We can also create an application to deploy this model on a user-friendly website where it can be publicly available for access and anyone can upload the document from which they need to generate questions, select the type of question and then with a single click of a button be able to download or generate a list of questions along with the answers

Acknowledgement
We, students of REVA UNIVERSITY, would like to express our gratitude towards our Hon'ble Chancellor, Dr. P. Shyama Raju, Hon'ble Vice-Chancellor Dr. M Dhanamjaya, for providing us the opportunity to enhance our knowledge in an environment that promotes growth and innovation. We would also like to thank our respected Director and Project Guide, Dr. Sunil Kumar S. Manvi, for his constant support and guidance throughout the project. We are also thankful to all faculty members for their kind contributions.

References:

[1] Xinya Du, Junru Shao, Claire Cardie : "Learning to Ask: Neural Question Generation for Reading Comprehension" in arXiv:1705.00106v1 [cs.CL], 29 Apr 2017.
[2] Lior Sidi and Hadar Klein : "Neural Network-Based Collaborative Filtering for Question Sequencing" in arXiv:2004.12212v1 [cs.LG], 25 Apr 2020.
[3] Leonid Velikovich, Ian Williams, Justin Scheiner, Petar Aleksic, Pedro Moreno, Michael Riley: "Semantic Lattice Processing in Contextual Automatic Speech Recognition for Google Assistant" in Proc. ISCA, 2018.
[4] Assaf Hurwitz, Michaely, Xuedong Zhang, Gabor Simko, Carolina Parada, Petar Aleksic: "Keyword Spotting for Google Assistant Using Contextual Speech Recognition" in Proc. IEEE ASRU, 2017.
[5] Michael Heilman, Noah A. Smith : "Good Question! Statistical Ranking for Question Generation" in N10-1086, Pg. 609-617, June 2010
[6] Dr.P Pabitha, M.Mohana, S.Suganthi, B.Sivanandhini :" Automatic Question Generation System" IEEE 2014 International Conference on Recent Trends in Information Technology
[7] A. Shirude, S. Totaia, S. Nikhar, Dr. V. Attar:"Automated Question Generation Tool for Structured Data" -2015 IEEE
[8] Akhil Killawala, Igor Khokhlov, Leon Reznik: "Computational Intelligence Framework for Automatic Quiz Question Generation" -2018 IEEE
[9] Hardik Mahipal Surana, Giri Bhatnagar, Akshay Srivastava, Gowri Srinivasa : "NVR Guess: Automated Question Generation for Honing NonVerbal Reasoning Skills" 2019 IEEE Tenth International Conference on Technology for Education (T4E)
[10] Animesh Srivastava, Shantanu Shinde, Naeem Patel,Siddhesh Despande,Anuj Dalvi,Shweta Tripathi: "Questionator - Automated Question Generation using Deep Learning" IEEE 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)
[11] Husam Ali, Yllias Chali ,Sadid A. Hasan - "Automatic Question Generation from Sentences" TALN 2010, Montréal
[12] Ashwin K Vijayakumar1 , Michael Cogswell1 , Ramprasath R. Selvaraju1 , Qing Sun1 Stefan Lee1 , David Crandall2 & Dhruv Batra1 - "Diverse Beam Search : Decoding diverse solutions from neural sequence models" -2018.
[13] Dongling Xiao , Han Zhang , Yukun Li , Yu Sun , Hao Tian , Hua Wu and Haifeng Wang: ERNIE-GEN: An Enhanced Multi-Flow Pre-training and Fine-tuning Framework for Natural Language Generation -Baidu Inc., China 8th June 2020
[14] Ashish Vaswani,Noam Shazeer,Niki Parmar,Jakob Uszkoreit,Llion Jones,Aidan N. Gomez,Łukasz Kaiser,Illia Polosukhin : "Attention is all you Need"-6th Dec 2017
[15] Christopher Clark, Kenton Lee, Ming-Wei Chang , Tom Kwiatkowski, Michael Collins, Kristina Toutanova:"BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions"- 24th May, 2019
[16] Sudhan Murugan Bhagavathi, Anitha Thavasimuthu, Aruna Murugesan, Charlyn Pushpa Latha George Rajendran, A Vijay, Raja Laxmi, Rajendran Thavasimuthu, Weather forecasting and prediction using hybrid C5.0 machine learning algorithm International Journal of Communication Systems, Vol. 34, Issue. 10, Pp. e4805, 2021.
[17] PM Surendra, S Manimurugan, A New Modified Recurrent Extreme Learning with PSO Machine Based on Feature Fusion with CNN Deep Features for Breast Cancer Detection, Journal of Computational Science and Intelligent Technologies, Vol. 1, Issue. 3, Pp. 15-21, 2020.
[18] PK Sadineni, Comparative Study on Query Processing and Indexing Techniques in Big Data, 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), pp. 933-939, 2020.
[19] AH Omar Baabood, Prajoona Valsalan, Tariq Ahmed Barham Baomar, IoT Based Health Monitoring System, Journal of Critical Reviews , Vol. 7, Issue. 4, pp. 739-743, 2020.
[20] Sajay KR, Suvanam Sasidhar Babu, Vijayalakshmi Yellepeddi, Enhancing The Security Of Cloud Data Using Hybrid Encryption Algorithm, Journal of Ambient Intelligence and Humanized Computing, 2019. https://doi.org/10.1007/s12652-019-01403-1
[21] Bindhia K Francis, Suvanam Sasidhar Babu, Predicting academic performance of students using a hybrid data mining approach, Journal of Medical Systems, 43:162, 2019. https://doi.org/10.1007/s10916-019-1295-4