Journal of Engineering Education Transformations ,
Volume 38 , No.1 , July 2024 , ISSN 2349-2473, eISSN 2394-1707

10.16920/jeet/2024/v38i1/24180

Qgen: A Unique Question Generation and Answer Evaluation Technique Using Natural Language Processing

Sumedh Vichare[1], Aruna Gawade[2], Ramchandra Mangrulkar[3]
[1,2,3] Dwarkadas J. Sanghvi College of Engineering, Mumbai, Maharashtra, India.
[1]sumedhuvichare@gmail.com,
[2]aruna.gawade@djsce.ac.in,
[3]ramchandra.mangrulkar@djsce.ac.in

Abstract: Educational infrastructure is moving towards rapid digitization to conduct and evaluate examinations for remote students. Many universities now offer globally recognized distance learning courses to cater to a wider audience. However, this transition comes with its set of challenges, particularly for professors and staff members who find themselves burdened with a substantial amount of manual work during the examination season. The tasks include setting up unique question papers for every exam, including different types of questions with varying difficulty, and eventually evaluating the answers given by the students, which is not only time-consuming but also a labour-intensive process.

To address this issue, the paper proposes a solution that aims to reduce the workload of teaching staff by enhancing the efficiency of the examination process. It does so by leveraging several natural language processing techniques for generating two types of questions- objective and subjective, and grading the solutions of the examinee. Additionally, subjective questions are further classified based on Bloom's taxonomy levels, providing a diverse range of questions that align with varying cognitive abilities. The automation of this process not only eases the burden on educators but also ensures a more streamlined and effective examination process, thus contributing to the broader goal of digitizing education.

Keywords: Automated Question Generation, Bloom's Taxonomy, Workload Reduction.

Sumedh Vichare
Dwarkadas J. Sanghvi College of Engineering, Mumbai, Maharashtra, India.
sumedhuvichare@gmail.com,

1. Introduction

With the onset of the pandemic, digital technologies have rapidly disrupted the educational sector. Traditional examinations were pen-and-paper-based tests that were held in examination halls under the supervision of invigilators. The seating arrangements for students and the issuance of hall tickets for exams play a crucial role in the successful conduct of such traditional examinations. However, there is a fair set of challenges that come with these traditional exams. They are listed as follows:

· High manpower is necessary to facilitate a good student-to-faculty ratio for the efficient conduction of all stages of the examination. It is labor-intensive and costly.

· A huge number of unique questions must be created manually by the teaching staff for each

examination to maintain the difficulty and standard of each exam.

· The process is inefficient because of the lack of digital means to assist the professors with their work.

The increase in digital infrastructure during and post-pandemic has made high-quality education accessible to anyone with an internet connection. Highly reputed universities offer distance learning programs so students can complete their entire degree remotely without even attending a single day of offline college. Examinations are digitally held with the use of platforms that allow professors to upload the question papers and allow students to upload their answers. They are usually conducted in centers that provide computers and internet connectivity. They are useful for conducting academic tests, entrance exams, interview aptitude tests, etc. Human involvement in areas like examination conduction and assignment submission has been eliminated because of this digitization.

In the study done by (Talib, Betayeb, & Omer, 2021), it was found that the COVID-19 pandemic has had a significant impact on the use of technology in higher education. Many institutions were forced to move to online learning overnight, which required them to rapidly adopt new technologies and teaching methods. This transition was not always smooth, and many students and faculty experienced challenges. However, the paper also found that the pandemic has also created some opportunities for technology-based education. It elaborates on how leveraging the potential of technology during this time allowed education to still be accessible to the masses. It suggests that online learning can make education more accessible to people from all walks of life. This could potentially lead to a more educated and skilled workforce, which could benefit society as a whole. Additionally, online learning can help to break down geographical barriers, which could lead to a more interconnected and globalized society.

But this also has its fair share of challenges. The rate of increase in the number of students has been far greater than the rate of increase in the teaching staff for online courses.

The ability to efficiently conduct remote online examinations manually and have sufficient unique questions has been a bottleneck for this ecosystem. Presently, even this infrastructure is heavily reliant on high manpower for its smooth functioning. This puts immense pressure on the teaching staff to not only spend time offering quality online educational lectures but also spend time on tasks such as creating a question set, grading the uploaded solution of students, and even proctoring them during exams.

In the study by (Pace, D'Urso, Zappulla, & Pace, 2021), the authors explored the relationship between workload and personal well-being among 252 university professors. The study revealed that when university professors face excessive bureaucracy, it can lead to a more negative perception of their work-related well-being due to increased workload.

Hence, an effective way of conducting examinations, generating questions, as well as auto-evaluating them online is necessary to make the whole process less burdensome and more efficient.

In this paper, a system is presented that will automatically validate a student's attendance during the examination by mapping the identity to the database. The students would then be given questions based on the question bank created using the question generation software in 7the system. This software makes use of natural language processing techniques such as word sense disambiguation and part-of-speech tagging to frame a meaningful Wh-type question. The objective, as well as subjective-based questions, will be automatically evaluated using the modal answer generated along with the questions, and the students will be instantly given their grades. This reduces the waiting time for students and the effort that goes into generating questions for the professor. A lot of time and resources are saved during this entire process.

Additionally, several potential benefits can be anticipated in terms of the learning outcome. Firstly, the time and energy saved from all the manual effort can be utilized towards more valuable tasks by the teaching staff such as refining teaching methodologies and conducting discussions through increased contact hours. This may enhance the learning experience of the students by giving them personalized support from the experts in their field since they're no longer occupied with administrative tasks. This shift of focus of professors into more pedagogical activities would lead to an improved quality of education.

Moreover, the use of NLP techniques for question

generation would lead to generations of more diverse and complex questions. When students encounter this wide range of questions, they are encouraged to analyze critically and apply their knowledge in more novel ways. This fosters a more engaging assessment process.

Lastly, the efficiency of this NLP-driven question generation streamlines and accelerates the entire examination process. Through feedback from students and educators, the intensity of examinations and complexity of questions can be calibrated to ensure maximum learning progress. It is more convenient to adjust teaching strategies because of the automated nature of examination conduction and evaluation.

2. Review Of Related Works

(Plisson et al., 2004) proposed a Rule-based approach to word Lemmatization. Answers that are typed by the student would be processed for identifying root words and finding the equivalent roots from the teacher's submitted answer. For example, “playing, plays, and played” would all be reduced to the root word called play. This will be done with the help of lemmatization. Lemmatization is an important pre-processing step for many applications of text mining. It is also used in natural language processing and many other fields that deal with linguistics in general. Based on the percentage of the match with the actual answer and the sample answer submitted by the teacher, the answers would be graded.

(Nenkova & McKeown, 2012) proposed a summarization technique that allows for a better comparison of the two texts. Students can intentionally overuse the keywords in the answer to get a higher score. For example, if the keyword is 'forest', an answer should not be graded highly if it includes 'forest forest forest', which does not make sense. To counter this problem, long texts must be summarized and then compared with the sample answers to be fairly graded.

(Porwal et al., 2022) devised an audio transcript generation technique. Online Vivas conducted in a real-time virtual meeting can be tedious for a professor with a large number of students. There is also little to no scope for efficient proctoring under such time constraints. Students will be prompted with a question and will have some time to read the question after which the recording will automatically start. In case of any contact with the keyboard or change of tabs while answering, it will auto-submit the viva, and a malpractice case will be reported.

(Lakshmi & Ramesh, 2017) discussed various preprocessing and processing techniques to generate questions based on input passage. The paper achieved an accuracy of 71 percent through this technique. The use of libraries such as NLTK and spaCy in Python to generate questions has been suggested in this paper. The paper describes various preprocessing and sentence selection techniques for achieving a higher quality of questions.

(Joshi et al., n.d.) provided a natural language processing-based approach for evaluating response scripts by an algorithm. To rate the answer script, text from the answer is extracted, the recovered text is compared to the stored right answers to compute similarity, and a weight value is applied to each measure. The obtained information is then used to create a summary using keyword-based summarizing algorithms. Four similarity measurements (Cosine) are used as parameters to create the final mark. These studies have demonstrated the value of automatic evaluation of answer scripts and the consistency between the supplied scores and the hand-marked scores.

(S. F. Kusuma et al., 2018) discuss the three core components of a multiple-choice question, namely the distractor, the stem, and the key. It emphasizes the importance of a good distractor set to prepare high-quality questions. The paper also discusses and reviews at length the various phases of MCQ generation and the number of techniques for each of those phases. The phases are pre-processing, sentence selection, key selection, question-formation, distractor selection, and post-processing. 86 articles have been analyzed, and the findings have been reviewed in this paper.

(Sinha et al., 2022) developed a question generation system consisting of two modules: content selection and question formation. Content selection involves identifying the appropriate text section, while question formation includes disambiguating connectives, determining question type, and applying syntactic transformations. The researcher examines seven connectives, including "because," "since," "as a result," "for example," and "on that basis." Question type is determined based on these connectives; for

example, "since" corresponds to "Why." Two evaluators assess question accuracy.

In the paper by (Narendra et al., 2013), a system was developed where users input a text file to retrieve questions categorized by Bloom's taxonomy. Generating questions aligned with Bloom's taxonomy effectively assesses learning abilities. The framework employs agents for document processing, information classification, and question generation, creating a multi-agent system.

To automate the process, the system uses a document processing tool and stemming, eliminating human intervention. Information categorization involves analyzing keywords generated through data processing and determining Bloom's category by searching for an appropriate action verb.

The question generation module constructs questions using a template-based strategy based on information classification results. This strategy matches selected keywords with relevant Bloom's levels.

(Agarwal et al., 2011) developed a system that generates targeted questions to support students' learning during the writing process. A case study involved 24 human supervisors and 33 research students. Questions generated by G-Asks were compared with those generated by humans. The authors analyzed prevalent question types derived from human inquiries and explained the question development process based on the original text. Evaluation includes precision, recall, and Cohen's Kappa coefficient for comparison, citation classification, and question quality.

In (Pandey & Rajeswari, 2013), the system selects the informative sentence and keyword based on semantic labels and named entities. Distractors are chosen using sentence similarity. Questions are generated in the form of multiple-choice questions about a word in a given sentence, such as an adjective, adverb, or vocabulary term. Semantic Role Labeler and NER (Named Entity Recognizer) are used to generate questions and identify names, locations, or organization names. Similarity between the question sentence and sentences in the question knowledge base is measured.

Additionally, (Liu et al., 2012) employ semantic labels and named entities to select the informative sentence and keyword. Distractors are chosen based on sentence similarity. The study focuses on generating automatic multiple-choice questions about a specific word in a sentence, such as an adjective, adverb, or vocabulary term. Semantic Role Labeler and NER are used to generate questions and identify names, locations, or organization names. Similarity between the question sentence and the question knowledge base is assessed.

However, (Fattoh, 2014) uses Semantic Role Labelling and Named Entity Recognizer (NER) to transform the given sentence into a semantic pattern. An artificial immune system is developed to classify patterns by question type, particularly WH-questions (who, when, where, why, how). The immune system includes feature extraction, learning, memory storage, and associative retrieval to address recognition and classification challenges. NER and SRL techniques parse the input sentence to determine if it contains a person's name, location, or date, which determines the question pattern.

(Uto & Uchida, 2020) proposed a Deep Neural Network (DNN) based approach for automated short answer grading (ASAG). It uses an item response theory (IRT) model to estimate the test taker's ability through the objective true/false questions solved on the same test. The short answer grade is then evaluated by using a combination of the test-taker's ability along with distributed short

3. Methodology

Subjective questions are formed such that the answer typically involves more than 2-3 sentences. The phases of generating such a question are discussed in the following subsections.

Subjective Question Generation

Algorithm 1: Text Preprocessing

1. Input: Input Text for generating questions

2. IF: the input passage exists, perform the following steps:

a. Break down each sentence into smaller units using tokenization.

b. Lowercase each independent token.

c. Apply lemmatization using WordNet Lemmatizer in the Python NLTK library.

3. ELSE: prompt the user to input a sentence.

4. Output: Tokenized and lowercase words.

Consider a sentence:

- It is a good day.

After preprocessing: 'it', 'is', 'a', 'good', 'day'.

Reason for choosing Lemmatization over Stemming:

Both lemmatization and stemming techniques work towards reducing various forms of the same word into their root word. For instance, the words 'changing, changes, changed, changer' are all reduced to their root word 'change' in lemmatization. However, stemming would reduce these words to 'chang'. This makes stemming a good choice for use cases such as spam classification in email. While dealing with academic texts, preserving the meaning of the word is important. Hence, lemmatization provides better results in this system.

Algorithm 2: Text Processing

1. Input: Pre-processed text

2. If the input passage exists, perform the following steps:

a. Identify discourse markers using rule-based regex matching in Python.

b. Identify the question and answer part.

c. Perform parts of speech tagging on the question part using the NLTK library in Python.

3. Else prompt the user to input a sentence.

4. Output: Part of speech tagging on each token (word).

Discourse relations are shown using discourse connectives, which are words or phrases that link or relate two coherent sentences or phrases. Identifying discourse connectives helps identify two potentially interconnected parts of the same sentence. On identifying the specific type of the discourse marker based on the table, one part of the sentence is used for question generation and the second part could be the answer. After this, Part of Speech (POS) tagging is done to identify all the Noun-Verb combinations. POS tagging is done to understand the ambiguity of a word depending on the part of speech it belongs to.

Algorithm 3: Question Generation

1. Input: Processed text

a. If the input passage exists, perform the following steps:

b. Identify the verb based on the Noun-Verb combination using Part of Speech Tagging.

c. Add an auxiliary verb.

d. Merge the sentence to make a question.

e. Find the wh-word.

f. Form the question.

2. If the input passage does not exist, prompt the user to input a sentence.

3. Output: A Wh-type question.

Based on the POS tagging, an auxiliary verb is generated that is appropriate for the question formation. A suitable wh-word is then decided based on the specific tagged words in the sentence. Once this is done, the wh-word is placed at the start of the sentence along with the auxiliary verb, and the question is constructed.

Consider the sentence- 'The man sits under the tree because he is hurt.'

The discourse marker 'because' is identified. The appropriate wh-question for 'because' belongs to the causal category, and hence 'why' is chosen. The question is formed as 'Why does the man sit under the tree?' with the auxiliary verb 'does'.

POS tagging is broadly classified into two types. Rule-based and Statistical-based.

Rule-based tagging relies on predefined grammatical rules and patterns to assign POS tags to words. These rules are typically developed by

linguists or language experts based on linguistic knowledge and analysis. Rule-based tagging systems often use handcrafted lexical and syntactic rules to determine the POS tags of words. Examples of rule-based POS taggers include Brill's tagger and regular expression-based taggers.

Statistical or probabilistic approaches use statistical models and machine learning algorithms to determine the most likely POS tags for words. These models are trained on large annotated corpora, where words are tagged with their correct POS tags. Statistical models calculate the probability of a word having a specific POS tag based on the context of the word and surrounding words. Statistical taggers learn from data and can handle ambiguous cases by considering the statistical likelihood of different POS tag assignments. Hidden Markov Models (HMM), Maximum Entropy Markov Models (MEMM), and Conditional Random Fields (CRF) are commonly used Statistical models for POS tagging.

Python's NLTK library has a default rule-based POS tagger. It also supports stochastic tagging which uses probabilistic models trained on large amounts of annotated data to predict the most likely POS tags for words. The Averaged Perceptron Tagger in NLTK is a statistical part-of-speech (POS) tagger used for processing text in QGen.

The averaged perceptron tagger is trained on a large corpus of text, which makes it more robust and accurate than the default rule-based tagger provided by NLTK. It also allows you to specify the tagset, which is the set of POS tags that can be used for tagging; in this case, it's using the 'universal' tagset, which is a cross-lingual tagset, useful for many NLP tasks in Python.

There are two approaches to Discourse Marker Identification-

1. Simple Approach

By creating a list of known discourse markers, we can use a simple rule-based approach to check if a token in the text matches any of the known discourse markers.

2. Advanced Approach

In Qgen, we've used an advanced approach for more accurate identification of discourse markers.

This involves training a model on a labeled dataset of text with annotated discourse markers, by using the spaCy library.

Bloom's Taxonomy is a framework that helps classify the cognitive abilities of the test taker into various sections starting from lower-order thinking skills to higher-order thinking skills. These levels are Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation. (Tabrizi & Rideout, 2017) discuss critical pedagogy, which focuses on empowering students through active learning by engaging students in higher-order thinking skills, and how incorporating Bloom's taxonomy helps identify and promotes an open form of creative thinking and questioning among students. Furthermore, a well-structured examination that includes questions from various Bloom's Taxonomy levels enhances the validity (measuring what it's intended to measure) and reliability (consistency of results) of the assessment. This makes the results more dependable and meaningful.

Objective Question Generation

Question types such as Multiple-choice questions, fill-in-the-blanks, and matching the following, true or false are generated in this section.

A. MCQ Generation

T5 transformer

The T5 transformer model is a natural language processing (NLP) model developed by Google's AI research team. T5 stands for Text-To-Text Transfer Transformer, which refers to the fact that the model is trained to convert one textual input into another textual output. The T5 model is based on the Transformer architecture, which was introduced in a seminal paper by (Vaswani et al., 2017).

The T5 model is trained on a wide range of natural language tasks, including text summarization, machine translation, question answering, and sentence classification, among others. It is trained in a "pre-training and fine-tuning" paradigm, where it is first trained on a large corpus of text data in an

unsupervised manner, and then fine-tuned on specific downstream tasks with labeled data.

The T5 model has achieved state-of-the-art performance on several benchmark NLP tasks, and it is widely used in industry and academia for a variety of NLP applications.

For this system, it is trained on Stanford's SQuAD(Stanford Question Answering Dataset). The dataset comprises a question, an answer, and a context to the answer. On training this model, it is possible to provide an answer and a context to generate a multiple-choice question accurately.

Algorithm 4: MCQ Question Generation

1. Input: Processed text

   If the input passage exists, perform the following steps:

   a. Summarise the text.

   b. Use the T5 transformer model to generate the question.

2. If the input passage does not exist, prompt the user to input a sentence.

3. Output: A MCQ question.

Natural language processing methods are used in abstractive summarizing to identify the key ideas in a document, comprehend those ideas, and provide an appropriate summary. Using the T5(text-to-text transfer transformer) model, the question is generated and the correct answer is identified and stored in the system. It uses a transfer learning approach. Transfer learning, which involves pre-training a model on a task with plenty of data before fine-tuning it on a subsequent task, has become a potent method for natural language processing. A single model of T5 can do multiple tasks such as translation and summarization. The question is generated by following the steps mentioned in Algorithm 4.

Word Sense Disambiguation

It is a technique for identifying the context or meaning of the word used in a specific sentence or phrase. It is a process that occurs naturally in human beings but has to be understood in machines. In Python, WordNet bindings in the NLTK library are used to generate the most probable sense of a word. E.g. Consider the sentence

- The industrial plant consumes a lot of energy
- It is healthy to have a plant-based diet
- In both of these sentences, the context of the word plant differs which will be detected by the WordNet library in python

The two approaches to Word Sense Disambiguation have been outlined below.

1. Dictionary-based approach- Lesk algorithm.

As the name suggests, for disambiguation, these methods primarily rely on dictionaries, treasures, and lexical knowledge bases. They do not use corpora evidence for disambiguation. The Lesk method is the seminal dictionary-based method introduced by Michael Lesk in 1986. The Lesk definition, on which the Lesk algorithm is based is “measure overlap between sense definitions for all words in context”.

2. Machine learning-based approach

Supervised methods rely on sense-annotated corpora for training, assuming that context alone provides sufficient evidence for disambiguation without relying on explicit knowledge or reasoning. Context is represented as word features, including information from surrounding words, with support vector machines and memory-based learning being effective techniques. However, these methods require a substantial amount of manually annotated data, making them expensive to implement.

Semi-supervised approaches are employed when training corpora are limited, using a combination of labeled and unlabelled data. These methods leverage bootstrapping algorithms, which start with initial seed data and iteratively expand the list of things to disambiguate by incorporating information from a training corpus. This approach requires a small amount of annotated text and a large amount of unannotated text to effectively disambiguate word senses.

Unsupervised methods assume that similar word senses occur in similar contexts, allowing senses to be induced from the text by clustering word occurrences

based on contextual similarity. This process, known as word sense induction or discrimination, has the advantage of not depending on manual efforts for knowledge acquisition, making it a promising approach to overcoming data acquisition challenges in WSD.

In Qgen, the dictionary-based approach has been preferred provided by the NLTK library in Python, because of its optimum speed for processing large texts. NLTK also provides access to WordNet, a lexical database that includes word senses and their relationships.

Hypernym and Hyponym

Once we have identified the correct context of the answer in the sentence using word sense disambiguation, the next step is to generate incorrect options that closely resemble the correct answer. For this, we need to find the “umbrella” term for the solution word i.e. the Hypernym. Once the hypernym has been identified, all the words that fall under this umbrella term can be found. These words are known as Hyponyms. Two or more words belonging under the same Hypernym are known as co-hyponyms. As shown in Fig. 1, these co-hyponyms will be used as the incorrect options(distractors).

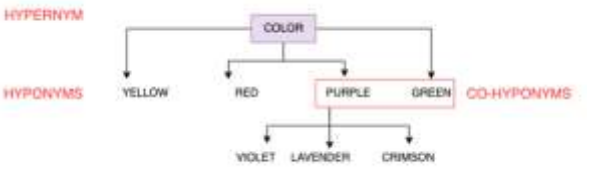

Fig. 1 : Relationship between Hypernyms and Hyponyms illustrated

A word can mean different things depending on the context of its usage. Word sense disambiguation is used to correctly identify the context of the correct answer in a given sentence. Wordnet is used to capture relations. A hypernym refers to an umbrella term or a blanket term for a group of hyponyms which consists of the correct answer. For example, "red" is a hyponym under the hypernym "color". To generate multiple incorrect options for the question, it is necessary to extract as many co-hyponyms to act as istractors from the correct answer.

B. Fill in the Blanks Question

In a passage, the keywords are first extracted which will be the blanks or the solution of the question. These keywords are extracted using Python's keyword extraction library. The original sentences are then reconstructed by replacing the keyword with a blank.

C. Match the following Question

Like the Fill in the Blanks question, the key phrases are extracted using a keyword extraction library in Python. The keywords are then mapped to their meanings from the passage and then the order of the keyword and its meanings are jumbled to generate the question.

D. True or False Question

For this category of questions, there are two main stages involved. Firstly, statements that are objectively true as per the given input text must be created. This can be done by splitting a compound or complex sentence present in the passage into a simple sentence. Next, falsified sentences need to be generated.

Common approaches for falsifying a sentence

Negate or remove the negation of a verb phrase or noun phrase. For example, consider the sentence 'Jack doesn't swim'. This sentence can be altered to 'Jack swims' or vice-versa.

- Changing a named entity. For example, changing 'Mercury is the first planet in the solar system' to 'Venus is the first planet in the solar system'

- Changing the adjective. For example, 'Thanos is the scariest villain' will be converted to 'Thanos is the softest villain'.

- Changing the main verb. For example, 'When electrons are shared, a covalent bond is formed' will be converted to 'When electrons are transferred, covalent bonds are formed'.

A better approach for falsifying a sentence

The methods covered in the previous section are restricted to the occurrence of certain nouns or adjectives to get the desired result but we need a more generalized approach for a huge data set. For this, the sentence is parsed using constituency parsing. As shown in Fig. 2, it is a technique where sentences are broken down into sub-phrases(constituents). It is

Fig. 2: Sentence constituency parsing for splitting the ending noun or verb phrase.

Fig. 3 : Illustrating how falsified sentences are generated using GPT-2 auto-completion.

stored in the form of a tree data structure where the root is the sentence and the leaf nodes are the words. The noun or verb phrase is replaced using the sentence auto-complete feature of open AI GPT-2 which has been shown in Fig. 3. The completed sentences are then compared with the original sentence using cosine similarity to select the most appropriate and contextually correct false statement. This is a more generalized approach that works for a wide range of sentences and yields better accuracy. The covers the sequence of processes to generate the falsified sentence.

Evaluating Answers

At the time of question generation, a question-answer pair is created. The answer entered by the student is compared with the model answer after the removal of stop words using cosine similarity to generate a score. Feedback generated by the transformer model is also displayed when an incorrect answer is given.

4. System Architecture

Fig. 4. illustrates the system architecture comprising three primary actors.

- Teaching Staff
- Admin Staff
- Student

The teaching staff has access to all the relevant course materials which will be used for question generation. The staff must hence create a database of all student data and generate an appropriate number of

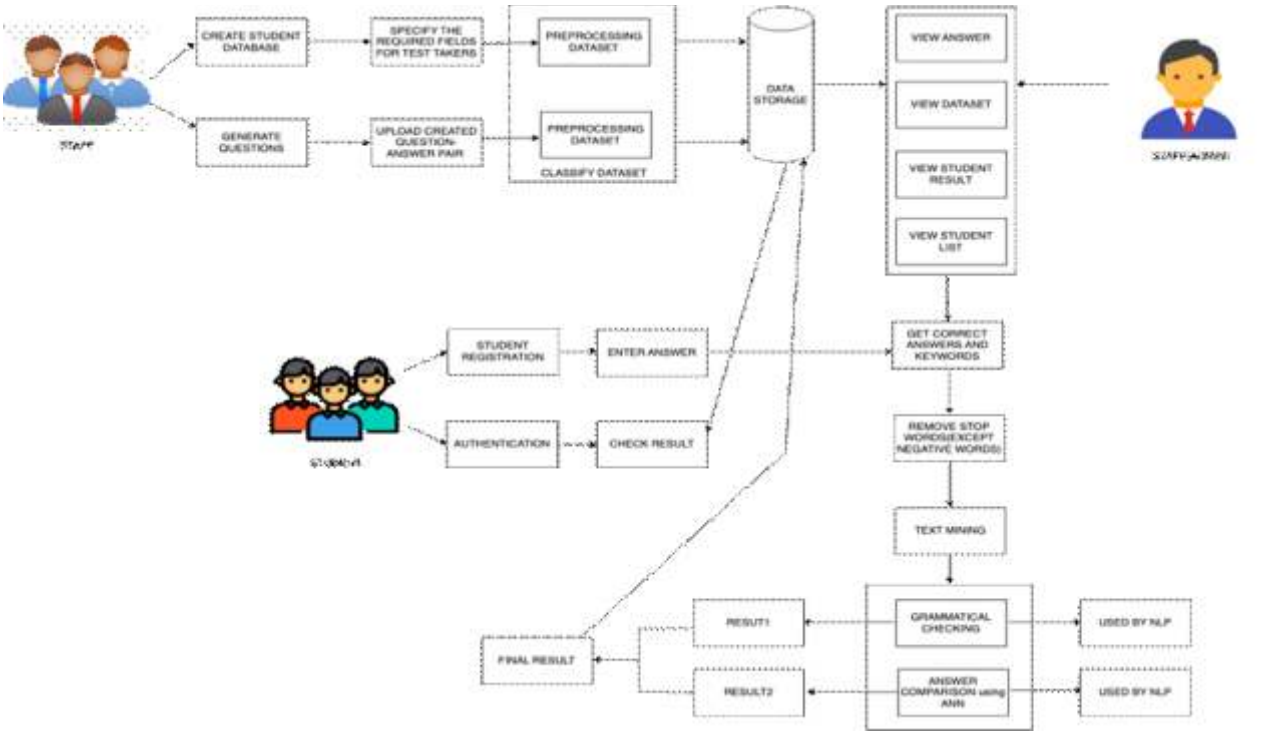

Fig. 4 : System architecture outlining the roles of all three users.

questions for the specified students. All of this data will be then stored in the database. For question generation, the passage from the respective curriculum in the exam needs to be entered by the staff. This data will be pre-processed and then stored in the database. The students have two types of tasks. Firstly, the students must authenticate themselves through the portal. This will be facial recognition-based. After this, the student will be directed to the examination window. The examination window consists of questions that have to be answered by the students. For objective-type questions, the student must select the correct options. A timer will be running on the right-hand corner of the screen. On successful submission of answers by the students, they will be redirected to the subjective question screen. The answers to these questions must be typed by the students within the stipulated word count limit. On successful completion of the task, the examination session will end. Now, the students can log in to their reports section and view the scores for both their subjective and objective answers. The admin staff has the task of maintaining and managing all the student data along with the question-answer pair data generated by the teaching staff. They will also have access to all the results of all the students who have taken the test. In case of any discrepancy, the changes or modifications on student data or examination data will be carried out manually by the admin staff.

QGen System:

All the features discussed in this paper in terms of examination conduction is implemented into a single system with an interface for professor as well as student.

A. Staff (Professor) Interface

- A software interface for professors to enter a sample passage for which a question bank needs to be generated.
- The text will be pre-processed using tokenization, lower-casing, and lemmatization.
- Sentence processing will contain POS Tagging, discourse marker identifier, and generating an auxiliary verb depending on the Noun-Verb combination.
- Merging the sentence to generate a question.
- Considering the original sentence as the model answer.
- After the student inputs the solution, check it for grammatical correctness and remove the stop words.
- Checking the cosine similarity of the student's answer with the model answer and checking for keyword synonyms match percentage to calculate the final score.
- Validate the student's identity at the beginning of the test.

On logging into the system, the professor will be required to enter the text passage for which the questions need to be generated. On entering the passage, options for generating either subjective or objective questions are given to the user along with the number of questions. The input passage needs to be of sufficient length corresponding to the number of questions. For instance, a 100-word passage might not be suitable to generate a question set of 20 unique questions. The UI puts a limit on the number of questions possible to generate based on the input number word count. The generated questions are then displayed and the professor has the option to export it as a CSV or any desired format.

B. Student Interface

Once students have entered their login credentials, their face is authenticated using a face-recognition library in Python. After the student's face is authenticated, the examination session will start. The students will then have to type various kinds of subjective questions given to them. On successful completion of the exam, the student can submit the answer paper. If time runs out, the answers will be auto-submitted. This answer would then be compared with the model answers created at the time of question generation and a computer-generated score would be given to the student for these answers.

C. Staff (Admin) Interface

This is an administrative interface to ensure the consistency of student information in case of any technical issues faced by the users of Qgen. Since the admin staff is the point of contact for students if anything goes wrong with their examination session, this interface is required to view and rectify inconsistencies.

5. Discussion On Existing Systems

(Chen, Yang, & Dragan Gašević, 2019) conducted a comprehensive analysis of nine sentence selection strategies inspired by various question-asking heuristics, emphasizing the significance of sentence selection for generating educational questions, an aspect often overlooked. Their extensive experimentation yielded consistent results, showing that LexRank, a method based on eigenvector centrality for identifying important sentences within articles, maintained strong performance across various datasets. Additionally, their study illuminated the contrast between sentence selection in non-educational and educational contexts. In the former, preference was often given to the opening sentence of an article, while the latter demanded a more diverse range of informative, vital, or innovative source sentences.

LexRank is based on graph-based methods and involves calculating eigenvector centrality in a graph of sentence similarities. The time it takes to perform these calculations can increase with the length of the input text and the number of sentences being considered. In general, LexRank can be relatively time-consuming for very large texts or when dealing with a large number of sentences.

Qgen employs a keyword extraction approach using the transformer model for sentence selection which doesn't demand any complex computation and performs with reasonable accuracy for academic input texts. This is because the dataset used (SQuAD) for training the T5 transformer is content-agnostic (generic) and not content-specific, which allows for the high quality of questions being generated regardless of the domain.

(Tahani Alsubait, Bijan Parsia, & Sattler, 2015) stated that controlling the difficulty of the questions generated was a limitation in their review of 38 text-based approaches to question generation that were identified in their study. Qgen has addressed this limitation by incorporating Bloom's Levels for the Questions generated.

(Kurdi, Leo, Parsia, Sattler, & Al-Emari, 2019) reviewed 93 papers on Automated Question Generation (AQG), highlighting issues with question generation and evaluation. The study also discussed the significant role of template libraries in question generation systems, noting that the current manual template construction process is both time-consuming and resource-intensive. They emphasized the need for improving question quality, exploring higher-order questions, and automating template creation. The importance of natural language processing for correct question presentation was stressed, and the a need for richer feedback generation.

The major shortcomings of the papers on AGC between 2015 and early 2019 have been discussed in this study and Qgen addresses some of these major issues of generating different complexity of questions, test feedback, and using natural language processing for generating diverse questions effectively.

6. Results

The system authenticates students and allows them to get instant grades for their subjective answers. In this implementation, the subjective and multiple-choice questions are generated by the system. NLP libraries such as NLTK are used for performing semantic operations on the given input text for pre-processing and generating questions as required. The goal of this is to produce as many high-quality questions as possible from a given input passage. For each question, a question-answer pair will be created where the answer will be acting as the model answer. It is necessary to keep this in the database so the student's unique answer can be compared with the model answer for correctness and evaluation. The keywords and grammatical semantics along with cosine similarity are used to calculate the final grade of the student and then scaled to adjust the weightage of the question. The time taken to generate 4 questions from a passage of approximately 150 words is seen to be around 6.2 seconds.

The preliminary results indicate that for an examination lasting for 3 hours, the entire question paper set can be generated within 7-10 minutes to yield highly accurate questions for a total of 75-80 marks at the undergraduate level. This is more efficient than the typical examination setup which has limitations on the number of unique questions and consumes several hours of the teaching staff responsible for conducting the exam.

Qgen facilitates comprehensive evaluation by allowing educators to include questions that span various cognitive levels and learning domains. This means that assessments can go beyond rote memorization and test higher-order thinking skills

across different Bloom's levels such as analysis, synthesis, and evaluation. It enables a more holistic assessment of students' knowledge and abilities, contributing to a richer educational experience. There is added flexibility to modify the question types based on student feedback and this caters to the evolving needs of students. By embracing Qgen, educators and institutions can provide students with a more enriching and effective learning experience, ultimately contributing significantly to the broader field of educational technology and assessment.

Qgen generates questions based on the input passage after analyzing the context and nature of the text. It can be easily adapted to different areas of education that are outlined below.

History: Qgen can be used to generate questions that challenge students to analyze historical events from a different perspective. Since (McCullagh, 2000) talks about historical biases that exist in literature, it becomes crucial to compare contemporary interpretation with historical interpretation to effectively progress in this field while testing students' understanding of historical context and significance.

Psychology: Complex scenarios that test the students' understanding of psychological theories can be developed in Qgen. This bolsters the understanding of the research methodologies by creating examples of experimental design and statistical analysis. For example, questions that ask students to design an experiment to test a specific cognitive theory can integrate both theoretical and practical research skills.

Business: By using the questions generated by Bloom's levels, different questions could be used to generate case studies, simulations, and other interactive learning experiences. This would help students to develop their critical thinking and problem-solving skills.

Law: Questions about moot court and bar exams can be generated apart from the questions that require memorization. Qgen can generate questions on constitutional law where each question requires a different level of cognitive engagement – from remembering key principles to applying them in complex legal situations.

Healthcare: Questions can be developed using Qgen which simulates clinical scenarios where students must outline the plan of action and propose treatment plans for virtual patients. In a pharmacology course, objective-type questions that ask students to choose appropriate drug treatments based on the patient's history and current condition can foster a more engaging assessment technique.

Practical Implementation Aspects:

For generating multiple choice questions, we've made use of the SQuAD dataset in the T5 Transformer model. One challenge is that the SQuAD dataset is relatively small. This means that as the scope of this project extends beyond the realm of engineering and technology institutes, it may be difficult to generate high-quality questions from the dataset alone. It may be necessary to use a larger dataset or to pre-train the T5 transformer on a different dataset before fine-tuning it on the SQuAD dataset.

One challenge would be the collection of large sets of structured data to train the model across various fields of education. Once this has been achieved, it would be computationally expensive to train and run these language models. This can be a challenge for organizations that wish to conduct competitive exams on a large scale and do not have access to the necessary resources.

Finally, there is the challenge of ensuring the quality of the results. NLP techniques can sometimes generate erroneous results. Filtering these results on a large scale might require additional effort which could further increase time consumption. This can be a challenge for organizations that need to rely on the accuracy of the results.

7. Conclusion and Future Scope

In a comparison between conventional methods and Qgen, the time differences for various tasks are striking. Conventional blueprint creation takes 30 minutes, while Qgen does it in 30 seconds. Generating questions takes 2 hours conventionally, but only 10 minutes with Qgen. Solution grading with feedback, which consumes 3 hours traditionally, is reduced to a few minutes by Qgen. This results in a total time of 5 hours and 30 minutes for conventional methods versus roughly 11 minutes for Qgen, showcasing its remarkable efficiency and time-saving capabilities. Preparing a Blueprint refers to the classification of several questions, types of questions, difficulty, and weightage for each question to make up for the total

marks. Creating unique questions requires a subject matter expert (SME), usually a professor or teaching assistant in that area. Grading solution again requires a teaching or graduate assistant to help professors grade a vast number of students. The time taken for conventional way of exams has been averaged out over 100 responses from various engineering campus faculty across the city. Qgen reduces this time to just about 11 minutes and introduces a more dynamic set of questions overall. This research not only highlights the impressive efficiency gains of Qgen in comparison to conventional methods but also underscores its potential to transform the educational technology and assessment landscape. By embracing Qgen, educators, and institutions can save time, enhance assessment quality, and redirect resources towards improving the overall educational experience, making it a significant contribution to the broader field of educational technology and assessment.

Further research can be conducted to generate questions that can be used in real-time, such as during a lecture or a class discussion. This would help facilitate active learning and ensure that students are engaged with the material. Additionally, question generation can be carried out dynamically, one at a time, based on the student's performance during an exam. This approach would ensure adaptive difficulty levels for the test and dynamic scoring.

References

[1] Abu Talib, M., Bettayeb, A. M., & Omer, R. I. (2021). Analytical study on the impact of technology in higher education during the age of COVID-19: Systematic literature review. Education and Information Technologies, 26.

[2] Agarwal, M., Shah, R., & Mannem, P. (2011, June 1). Automatic Question Generation using Discourse Cues. ACLWeb; Association for Computational Linguistics.

[3] Chen, G., Yang, J., & Dragan Gašević. (2019). A Comparative Study on Question-Worthy Sentence Selection Strategies for Educational Question Generation. Lecture Notes in Computer Science, 59–70.

[4] Fattoh, I. E. (2014). Automatic Multiple Choice Question Generation System for Semantic Attributes Using String Similarity Measures. Computer Engineering and Intelligent Systems, 5(8), 66.

[5] Joshi, S., Shah, P., & Shah, S. (n.d.). Automatic question paper generation according to Bloom's taxonomy by generating questions from text using natural language processing| International Journal of Innovative Science and Research Technology.

[6] Kurdi, G., Leo, J., Parsia, B., Sattler, U., & Al-Emari, S. (2019). A Systematic Review of Automatic Question Generation for Educational Purposes. International Journal of Artificial Intelligence in Education, 30(1), 121–204.

[7] Lakshmi, V., & Ramesh, D. (2017). Evaluating Students' Descriptive Answers Using Natural Language Processing And Artificial Neural Networks. International Journal of Creative Research Thoughts - IJCRT, 5(Volume 5, Issue 4)

[8] Le, N.-T., Kojiri, T., & Pinkwart, N. (2014). Automatic Question Generation for Educational Applications – The State of Art. Advanced Computational Methods for Knowledge Engineering, 325–338.

[9] Liu, M., Calvo, R. A., & Rus, V. (2012). G-Asks: An Intelligent Automatic Question Generation System for Academic Writing Support. Dialogue & Discourse, 3(2), 101–124.

[10] Narendra, A., Agarwal, M., & Shah, R. (2013). Automatic Cloze-Questions Generation (pp. 7–13).

[11] Nenkova, A., & McKeown, K. (2012). A Survey of Text Summarization Techniques. In Mining Text Data (pp. 43–76).

[12] Pace, F., D'Urso, G., Zappulla, C., & Pace, U. (2019). The relation between workload and personal well-being among university professors. Current Psychology.

[13] Pandey, S., & Rajeswari, K. (2013). Automatic Question Generation Using Software Agents for Technical Institutions. In International Journal of Advanced Computer Research (pp.

2277–7970).

[14] Plisson, J., Lavrač, N., & Mladenić, D. (2004). A Rule-based Approach to Word Lemmatization. Semantic Scholar.

[15] Porwal, K., Srivastava, H., Gupta, R., Pratap Mall, S., & Gupta, N. (2022). Video Transcription and Summarization using NLP. SSRN Electronic Journal.

[16] Sinha, S. K., Yadav, S., & Verma, B. (2022, March 1). NLP-based Automatic Answer Evaluation. IEEE Xplore.

[17] S. F. Kusuma, R. Z. Alhamri, D. O. Siahaan, C. Fatichah and M. F. Naufal, "Indonesian Question Generation Based on Bloom's Taxonomy Using Text Analysis," 2018 International Seminar on Intelligent Technology and Its Applications (ISITIA), Bali, Indonesia, 2018, pp. 269-274

[18] Tabrizi, S., & Rideout, G. (2017). Active Learning: Using Bloom's Taxonomy to Support Critical Pedagogy. International Journal for Cross-Disciplinary Subjects in Education, 8(3), 3202–3209.

[19] Tahani Alsubait, Bijan Parsia, & Sattler, U. (2015). Generating Multiple Choice Questions From Ontologies: How Far Can We Go? Lecture Notes in Computer Science, 66–79.

[20] Uto, M., & Uchida, Y. (2020). Automated Short-Answer Grading Using Deep Neural Networks and Item Response Theory. Lecture Notes in Computer Science, 334–339.

[21] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention Is All You Need.

[22] Wang, Z., Valdez, J., Basu Mallick, D., & Baraniuk, R. G. (2022). Towards Human-Like Educational Question Generation with Large Language Models. Lecture Notes in Computer Science, 153–166.

[23] Mccullagh, C. B. (2000). Bias in Historical Description, Interpretation, and Explanation. History and Theory., 39(1), 39–66. https://doi.org/10.1111/0018-2656.00112