TEST AUTOMATIC GENERATION AN ALGORITHM FOR AN AUTOMATED TESTING SYSTEM

Roman Horbatiuk [1], Taras Sitkar [1], Roman Lutsyshyn [1], Stepan Sitkar [1] and Mykhailo Ozhha[1]

[1] *Ternopil Volodymyr Hnatiuk National Pedagogical University, Maksyma Kryvonosa street, 2, Ternopil, 46001, Ukraine*

Abstract

The main objective of this paper is to present the development of an automated testing system (ATS) for assessing and improving knowledge quality in educational institutions. The paper will explore the design, implementation, and assessment of the ATS, highlighting its significant features, benefits, limitations, and potential areas for further research. Furthermore, the paper will also discuss the system's impact on enhancing students' knowledge acquisition, retention, and application. Finally, the study will emphasize the importance of incorporating technology into the education sector to foster better learning outcomes, thus contributing to the ongoing discourse in the field of educational technology.

Keywords

Testing system, assessing knowledge quality, ATS, algorithm.

1. Introduction

An essential component of the educational system is evaluating students' knowledge. It aids teachers in figuring out how well students are picking up material and where they might need more assistance [1]. Additionally, it gives students feedback on their development, which is crucial for their growth. Written exams, projects, and presentations are just a few examples of the various assessment formats.

The use of formative assessment methods to improve student learning has received more attention in recent years [2]. Formative assessment is a continuous process that offers feedback to teachers and students frequently rather than just at the conclusion of a unit or course. There are many different ways to conduct formative assessments, including tests, surveys, and games [3]. These tests are frequently low-stakes, which means they don't factor into the student's final grade [4]. Instead, they give students immediate feedback on how well they comprehend a particular idea or subject. This feedback can then be used by the teacher to adjust their instruction and provide additional support to students who may be struggling [5]. Students who participate in formative assessments feel more progress and accomplishment in their learning.

One of the main advantages of formative assessment is that it enables teachers to spot potential problem areas in students at an early stage [6]. By doing this, teachers can keep their students from getting behind or losing motivation. Additionally, formative evaluations can encourage students to adopt a growth mindset because they realize that their knowledge and understanding can be improved with effort and practice rather than being fixed at a certain level [7].

Teachers can use a variety of formative assessment techniques in their classes. Exit tickets, for instance, are a quick and simple way to evaluate students' comprehension at the conclusion of a lesson.

Proceedings ITTAP'2023: 3rd International Workshop on Information Technologies: Theoretical and Applied Problems, November 22–24, 2023, Ternopil, Ukraine, Opole, Poland

EMAIL: gorbaroman@gmail.com (A. 1); sitkar@gmail.com (A. 2); lutsyshyn.ds@gmail.com (A. 3); sitkars@gmail.com (A. 4); misha.ochga@gmail.com (A. 5)

ORCID: 0000-0002-1497-1866 (A. 1); 0000-0002-5120-341X (A. 2); 0000-0002-3390-874X (A. 3); 0000-0003-4599-454X (A. 4); 0000-0002-6954-0318 (A. 5)

Students can be asked to write down one new concept they learned or one unanswered question they have about the subject. Future instruction can then be modified using this feedback [8]. Peer assessments, self-assessments, and reflective journals are additional examples of formative assessments [11].

Formative evaluations can be used to guide school-wide decision-making in addition to giving students and teachers feedback. Data from formative assessments, for instance, can be used to pinpoint areas where teachers or students might need more help or professional development [12]. The creation of school-wide initiatives to enhance student learning outcomes can then be guided by this information.

The use of formative assessments in the classroom comes with some difficulties. Making sure that the assessments are in line with the course's learning objectives and standards is a common challenge [13]. Additionally, designing and implementing formative assessments can take a lot of time, particularly if teachers employ a variety of methods. The teacher must also possess a certain level of expertise in order to interpret and effectively use the data from formative assessments in order to guide instruction [14].

Despite these difficulties, formative assessment is an effective method for fostering learning and participation among students in the classroom. Formative assessments help to make sure that all students have the support and resources they need to succeed by giving students and teachers continuous feedback. They are crucial to any successful educational system as a result.

While formative assessments have numerous advantages, they can be difficult for teachers to implement and manage effectively. Designing and administering assessments, for example, can be time-consuming, especially if teachers use a variety of techniques. Furthermore, collecting and analyzing data from formative assessments can be challenging, especially if teachers use paper-based methods. These difficulties can make it difficult for teachers to consistently use formative assessments and provide students with the feedback they require to succeed.

Many of these issues could be addressed with the help of an automated formative assessment system. A system like this could help teachers use formative assessment techniques more consistently by streamlining the process of designing and administering assessments [9]. Furthermore, an automated system could collect and analyze formative assessment data, providing teachers with valuable insights into student understanding and performance [10]. Teachers could save time and focus more on providing students with the support and feedback they require by automating many of the tasks associated with formative assessment.

Using an automated system for formative assessment has several potential advantages. An automated system, for example, could aid in the alignment of assessments with learning objectives and standards, making it easier for teachers to design effective assessments [16]. Furthermore, an automated system could provide students with immediate feedback on their understanding of a specific concept or topic, potentially increasing engagement and motivation [17]. Finally, an automated system could help to ensure that all students, regardless of teacher or classroom, have access to the same assessments and feedback.

However, there are some potential drawbacks to using an automated system for formative assessment. For example, ensuring that the system is accessible to all students, including those with disabilities or who do not have access to technology at home, may be difficult. Additionally, teachers may require training and support to effectively use the system, particularly if they are unfamiliar with technology-based assessment methods. Finally, there may be concerns about student data privacy and security, especially if the system is hosted by a third-party provider.

Despite these obstacles, an automated formative assessment system has the potential to be a useful tool for promoting student learning and engagement. A system like this could help to ensure that all students have the support and resources they need to succeed by streamlining the process of designing and administering assessments and providing immediate feedback to students. As such, it is a promising area for further research and development.

2. Development of an automated system for generating test tasks to assess the quality of knowledge

2.1. Description of the test task generation process

Neural networks are a type of artificial intelligence that is increasingly being used in educational settings. Neural networks are built to mimic the structure and function of the human brain, and they can learn from data and make predictions or classifications based on it. The feedforward neural network is one of the most common types of neural networks used in education.

Feedforward neural networks are designed to process input data through a series of layers, with each layer made up of a group of neurons that perform a specific function. The output of one layer is fed as input to the next layer, and the network's output is provided by the final layer. Feedforward neural networks can be used in education for a variety of tasks, including predicting student performance and identifying student misconceptions.

The recurrent neural network is another type of neural network that has been used in education. Recurrent neural networks are built to process sequential data like text or speech by allowing information to flow from one time step to the next. As a result, recurrent neural networks are wellsuited to tasks like language modeling and speech recognition.

Many potential applications for neural networks in education exist, including personalized learning, student performance prediction, and student modeling. A neural network, for example, could be trained on data from previous students to predict how well a current student will perform on a specific task. This data could then be used to personalize instruction for the student, as well as provide additional support if necessary.

Neural networks can also be used for student modeling, which is the process of creating a model of a student's knowledge and skills based on interactions with a learning system. This model can be used to provide personalized feedback and support to students in order to help them better understand the material.

Overall, neural networks have a wide range of potential applications in education, and their use is likely to grow as more research in this area is conducted. However, there are some drawbacks to using them, such as the need for large amounts of data and the possibility of bias in the data used to train the networks.

Many potential benefits of neural networks for assessing student knowledge include their ability to process large amounts of data quickly and accurately, learn from experience, and adapt to changing circumstances. The following are some of the specific benefits of using neural networks for student assessment:

1. Increased precision: Neural networks can make highly accurate predictions and classifications based on complex data sets. When compared to traditional assessment methods, this can result in more accurate assessments of student knowledge.

2. Personalized assessment: Because neural networks can be trained on data from individual students, more personalized assessments that account for each student's unique strengths and weaknesses are possible.

3. Immediate feedback: Neural networks can provide students with immediate feedback, allowing them to correct misconceptions and improve their understanding of the material faster.

4. Time-saving: Neural networks can process large amounts of data quickly, which can save time for teachers and other educators.

Despite these benefits, there are several drawbacks to using neural networks for student assessment. Among these limitations are:

1. Limited generalizability: Because neural networks are only as good as the data on which they are trained, they may not be effective in assessing students in contexts that differ significantly from the training data.

2. Potential bias: If the training data is biased, neural networks can be biased, leading to inaccurate assessments and reinforcing existing inequalities.

3. Lack of transparency: Because neural networks are difficult to interpret, it can be difficult to understand how they reach their conclusions and assess their reliability.

4. Technical requirements: Designing, training, and implementing neural networks requires significant technical expertise, which can be a barrier for some educators.

Overall, while neural networks have many potential advantages for assessing student knowledge, they must be carefully considered in terms of their limitations and potential biases before being used in educational settings.

Description of the data collection process

Any automated system for assessing student knowledge must include a data collection process. A neural network must be trained on a large dataset that is representative of the population it will be used to assess in order to accurately assess student knowledge. Several steps are involved in the data collection process, including:

Identifying the variables of interest: The first step in data collection is to identify the variables that will be used to assess student knowledge. These variables could include student demographics, prior academic performance, and assessment results.

Choosing the sample: After identifying the variables of interest, the next step is to choose a sample of students to participate in the study. The sample should be representative of the population being evaluated and large enough to allow the neural network to be trained on a diverse set of data.

Data collection: Once the sample has been chosen, data is collected using a variety of methods.

This could include giving tests, gathering information from online learning platforms, or conducting interviews or surveys.

Cleaning and organizing the data: After collecting the data, it must be cleaned and organized to ensure that it is accurate and usable. This may entail removing outliers or errors, converting data into a usable format, and labeling the data to indicate the correct answer or level of comprehension.

Splitting the data into training and testing sets: The final step in the data collection process is to divide the data into two groups: training and testing. The neural network is trained using the training set, and its performance is evaluated using the testing set. To avoid overfitting, ensure that the testing set is distinct from the training set.

Overall, data collection is an important step in creating an automated system for assessing student knowledge. Before training the neural network, it is critical to carefully consider the variables of interest, select a representative sample, and ensure that the data is accurate and usable.

Overview of the neural network architecture used for the system

The neural network architecture used in an automated system for assessing student knowledge can vary depending on the system's specific needs. However, several common components are typically included in the architecture, such as:

Input layer: The input layer is in charge of receiving data that will be used to evaluate student knowledge. Data on student demographics, academic performance, and previous assessment results could all be included.

Hidden layers: The neural network's hidden layers are in charge of processing input data and making predictions about student knowledge. The number of hidden layers and neurons in each layer can vary according to the difficulty of the problem being solved.

Activation functions: Activation functions are used to introduce nonlinearity into the neural network, allowing it to model complex relationships between input data and predicted output.

Output layer: The neural network's output layer is in charge of producing final predictions about student knowledge. Predictions about a student's understanding of specific concepts or overall performance on an assessment could be included.

Loss function: The loss function calculates the difference between the predicted and actual output. This enables the neural network to adjust its weights and biases over time in order to improve its predictions.

Optimization algorithm: The optimization algorithm is used during training to update the neural network's weights and biases in order to minimize the loss function. Overall, the architecture of the neural network used in an automated system for assessing student knowledge will be determined by the system's specific requirements. However, by incorporating input layers, hidden layers, activation functions, output layers, loss functions, and optimization algorithms, a neural network capable of accurately assessing student knowledge can be designed.

Explanation of the training and validation process

The training and validation process is an essential step in creating an automated system for assessing student knowledge using neural networks. The procedure entails training the neural network on a subset of the available data and validating its performance on another subset of the data. This procedure ensures that the neural network can generalize to new data and is not overfitting to the training data.

The neural network is trained by feeding it input data and the corresponding output labels, which could be the correct answer or a level of understanding. Based on the input data and the current values

of its weights and biases, the neural network makes predictions. A loss function, such as mean squared error or cross-entropy loss, is used to compare these predictions to the actual output labels.

The optimization algorithm is then used to adjust the weights and biases to minimize the loss function and improve prediction accuracy.

It is critical to monitor the neural network's performance on a separate validation set during the training process. This set of data is not used for training, but rather to evaluate the neural network's performance during the training process. This assists in determining whether the model is overfitting to the training data, which means it is learning specific examples from the training set rather than general concepts that could be applied to other unseen examples.

Validation metrics such as accuracy, precision, recall, and F1 score are used to assess the neural network's performance on the validation set. If the neural network's performance on the validation set does not improve, it may be necessary to adjust the neural network's hyperparameters, such as the learning rate or the number of hidden layers, or to restructure the neural network.

Overall, the training and validation process is essential for creating an accurate and dependable automated system for assessing student knowledge. It is possible to develop a system that can accurately assess student knowledge in a variety of contexts by monitoring the neural network's performance during the training process and adjusting its structure and hyperparameters as needed.

2.2. Automatic Generation of Multiple Choice Questions

We initiated the creation of a system that incorporates computational tools for producing MCQs from English texts automatically. This method would result in less complexity and time, as it can generate questions without any additional modifications.

We drew connections between nonidentical approaches and methods from the studied literature for the task of Question Generation in English. In an early stage of the work, with the intention of testing some of those resources as useful to evolve the design and determine if they were feasible for unborn work (a primary trial was conducted after exploring various techniques in affiliated workshops) we developed a workflow that is inspired by what we have learned and can be applied to our intended goal.

The system channel, which is divided into five steps: Pre-processing, Answer election, Question Generation, and Distractor Selection is shown in Figure 1. Essentially, Pre–Processing will prepare the textbook for the coming way, while in Revealed (as opposed to QPS) Election of rejoinder campaigners from the same textbook will serve as the foundation for generating questions using the materials.

As its name implies, Question Generation is made up of the styles that produce the textbook of questions and stems. The process of question generation follows an iterative block called rejoinder election and continues until the rejointer is elected. This means that the system generates practical questions only and moves on to the coming rejoinded one. Finally, in Distractor election we use styles which elect seeker disclaimers and then select candidates who wish to return to their original methods.

There are two methods to ask questions, including banning colorful options that can be chosen by electing distractors but are not available to both, with the difference being in Question Generation. Although both methods are rooted in workshop practices, the reason for carrying them is not solely to compare them but also because of our control over their interpretation. In the rule- grounded path, we have complete control and regulation can change at any time. The other path is based on workshops that manipulate the use of ANNs and data structures.

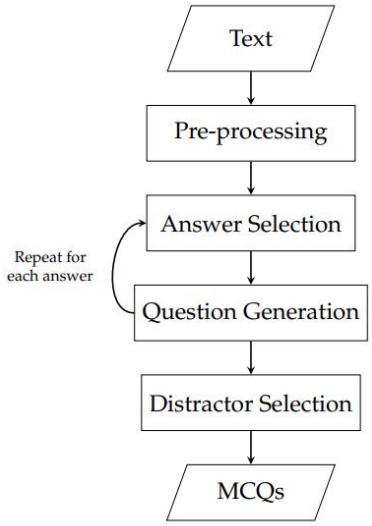

Figure 1. Diagram of a system operation.

Detailed explanations of the system steps are presented in Figure 1, along with some examples. The evaluation types and criteria chosen are also explained.

When in pre-processing, we can prepare the input written contents using the following approach: divide the textbook into lowerpieces if it is considered too long.

To deal with time or processing constraints, it may be feasible to sort the documents according to the applicability of rulings. Alternatively, summarization can alleviate this issue by keeping the main concepts in the textbook while making them less complex. This can be advantageous in practice.

Lemmatization and punctuation junking are two types of pre-processing methods that can be used to identify words in a document.

We focused on the resolution of co-references in our actions. Pronouns or expressions that make sense in written text are typically more complex in terms of context and textbook meaning because they relate to other expression types. When creating automatic questions, these same words can be included in similar questions too, making it difficult to fully grasp them.

To prepare the contents for subsequent stages of the channels, it is advisable to substitute these expressions with bones that correspond to them and are more easily identifiable than the originals. The neuralcoref library was used to perform co-reference resolution, which allows us to combine textbook sections that relate to the same thing and restore the entire textbook with these previously replaced expression patterns.

We used the "wikipedia" library to pre-save Wikipedia papers for trial and evaluation, which were also used as a source textbook and referencing tool for team dataset (Figure 2).

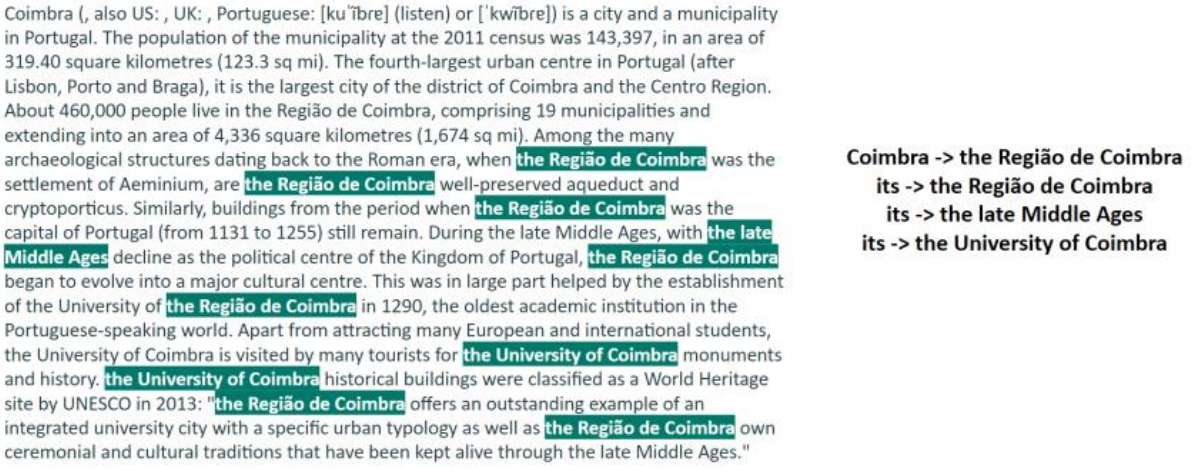

Figure 2. The Coimbra composition's co-reference resolution illustration is derived from Wikipedia.

Using the function that splits the document based on section titles, we loaded them into one of the papers. We made use of what had previously been textbook sections to divide the file by exploiting the newlines and leaving blank spaces for each section.

Process for determining an Answer Selection

The next step is to identify terms or expressions that could be used as answers after the textbook has been divided into lower documents and their content has already been processed. The system generates questions from answers because it requires the answers to be obtained first, similar to the Question Generation techniques of videlicet answer-apprehensive Transformers. To achieve this, we can automatically calculate on statistic or verbal information, such as TF- IDF, named realities, part-of-4 speech, etc.

Textbooks often contain primary generalities or ideas about certain motifs, particularly those that are more sphere-oriented. Word frequence can also serve as an indicator of usefulness by using it to validate the correctness of potentially non-applicable words like stop words and eliminate them.

The use of PoS trailing or shallow parsing can help identify the syntactic function of words, and we can also use words from a specific class (such as nouns or gobbets) as answer campaigners. Similarly, NER can be used to detect mentions of specific realities by identifying them. The type of reality can then determine how the judgment will be transformed into 'a question.'

We tested different styles, clauses, bigrams (single words), trigrams and combinations of terms, named realities, and noun gobbets in our development. We also experimented with Transformers for this purpose. The answer selection section was insulated from existing channels for question generation.

The final interpretation, based on the estimates made using all of these styles, includes named realities, noun gobbets, and campaigners named by a motor.

Our actions involve analyzing each choice in isolation while searching for answer campaigners. If there are specific realities, we employ SpaCy to identify these individuals and their reality markers.

The process is similar to that of noun gobbets, but with the added feature of assigning a reality marker to each no unto.

Our transformer method is based on the previously prepend question generation channel, where we isolate the section of the channel that selects answers that are grounded in a given environment and assign corresponding reality markers. This is similar to the way we assigned them for noun gobbets.

Question Generation

Once a seeker answer and the appropriate environment (such as judgment) are present, the objective is to induce the textbook of the question, which means that the stem or channel was designed to generate potential questions and follow up with candidates.

In some systems, the term in the judgment that will be the answer is replaced with a blank space. This ensures that the judgement remains declarative. However, we decided in advance that our system should transform the verdict into 'a question' as it sounded interesting and would add value to the work. Rule- grounded approaches can modify and restructure rulings by transfiguring their words. These changes must align with pre-existing rules, which are usually handcrafted, such as metamorphoses on subject- verbal cuemas.

An alternative approach could be the use of templates, which are typically simpler models that generate questions by adding a specific element from the original document to an almost ready-to-use question. However, this method requires much more hand-crafted templates than rules, making it less self-regulating.

The use of Transformers involves following the same workflow as previously mentioned. They are considered state-of-the-art models and can be easily acquired through the Hugging Face website. However, the lower control they possess is a major drawback to the game.

The proposition involves generating questions for each answer, inculcating questions to prompt them during development, but we aim to create questions that are relevant to all answers. However, similar to pre-processing, we conduct research on source textbooks by section and then perform the same process for other sections.

The imposed styles were established on rules and Transformers. While we utilized the previously OK-adapted system for this objective, we fully implemented the rule-based system, with the exception of libraries used for verbal analysis such as chancing clauses and named realities.

Mechanics: Transformer

In the early stages of the project, we conducted preliminary experimentation to explore potential resources for future work and test their feasibility. The transformer was used as a model based on T5 and was fine-tuned with SQuAD v1.1 for AQG during this testing.

Following that, we tested out more from a Github repository. The repository contains transformers that are answer-aware and require answers to be generated, such as QG, QA-QG or QGS Prepend, along with an E2E response generator. These transformer types are also based on the T5 model and trained on SQuAD v1. Additionally, they select answers, while QT is dedicated to Question Generation.

The system's final version is based on [20], which was the more effective one. However, we encountered difficulties in isolating the Question Generation segment to avoid deteriorating results when using other Answer Selection Methods. Additionally, they recommend using the answer-agnostic transformer (E2E).

An illustration is an answer-agnostic transformer (E2E) that answers to this question: "What is the name of Portugal's city?" If we use "Portugal" instead, we get the correct answer.

Distractor Selection

We employed a BART transformer that was already fine-tuned and trained using the dataset RACE[21], as per the recommendations of [22].

The method of including both the question and answer in the input requires a maximum length of 1024 characters, which can result in multiple distractors.

We used a transformer that was based on [22], but the dataset used to train it contains declarative sentences and interrogative examples. Although our questions were not in the same style as those in RACE, there may be some less convincing explanations due to the lack of available models for this task.

- To give an example, we selected the opponents for the answer "Roger Taylor" and ended up with the following distractors:
- The person in question is identified as "John Deacon".
- The woman in question is referred to as "Queen".
- "Freddie Mercury"
- "Jack Deacon"
- The British musical group.

We describe the pipeline that was created to automatically generate MCQs for English language. The pipeline is broken down into different stages, including Pre-processing, Answer Selection, Question Generation, and Distractor Selection. We discuss how each one was developed, the methods used, etc.

We provide a detailed account of the evaluation carried out to compare the different methods and their conclusions. This evaluation involved both automatic evaluation and human opinion evaluation.

During our examination of related documents, we came across two distinct evaluation metrics: automatic and human-opinion: We describe the process of both types of evaluation, the metrics employed, results obtained, and the outcomes of the processes.

During the development and final stages of the project, it was necessary to evaluate the effectiveness of all considered approaches. The evaluation process was mainly used to determine which approaches were most effective. In the last phase of work, the evaluation was conducted to arrive at conclusions about the developed system's performance.

The project involved analyzing different answer selection methods and models of question generation through automatic evaluation, with the use of metrics from scientific literature reviewed (BLEU and ROUGE) and reference data like SQuAD1 being used to support this. Each passage contains an inventory of answers, their location on the paragraph, and the human-created question for each passage.

We utilized human opinions for the final assessment. This evaluation not only provides us with a means to determine system performance but also takes into account end users, who are the primary beneficiaries of the system. To address this issue, we used two different methods: 1) accessing data from forms distributed to project participants (IPN) and 2) using Mindflow to gather information on the

quality of generated questions and 3) testing which distractor selection method yielded more relevant results.

Automated evaluation can quickly arrive at conclusions about the performance of methods, which can aid in selecting the most effective ones. Furthermore, automatic evaluation is reproducible and assigns all methods to identical evaluation metrics and tasks. To determine the best answer selection methods for use, we tested:

- Unigrams are used to represent individual words in the context of terms.
- Bigrams are a set of two-word word combinations;
- A set of three words grouped together as trigrams;
- Terminology: Terms that are grouped together (e.g., individuals, geographical locations, dates, etc.);
- The noun and its corresponding words are what makes it a chunk.
- Whether they are part of a sentence or the entirety of an entire sentence, clauses can be used.
- The classification of T+NEs+NCs involves the inclusion of terms, named entities, and noun chunks.

The dev set of SQuAD v1.1 was used as the reference, which includes passages that contain answerable questions and their answers. Figure 3 shows a portion of the dataset that includes articles, questions from Wikipedia, and references to other topics created by humans. These answers include "Mediterranean" or "a Mediterranean climate", "What kind of climate does southern Australia have?"

All the methods mentioned previously were tested with and without stop words to determine their impact on Answer Selection (Table 1 and Table 2). The answers were then sorted using TF-IDF, and the average of the values for each paragraph was used to calculate the metrics.

{"context":"Southern California contains a Mediterranean climate, with infrequent rain and many sunny days. Summers are hot and dry, while winters are a bit warm or mild and wet. Serious rain can occur unusually. In the summers, temperature ranges are 90-60's while as winters are 70-50's, usually all of Southern California have Mediterranean climate. But snow is very rare in the Southwest of the state, it occurs on the Southeast of the state.",

"qas":[

{"answers":[{"answer_start":31, "text":"Mediterranean"},{"answer_start":29, "text":"a Mediterranean climate"}, {"answer_start":31, "text":"Mediterranean"}], "question":"What kind of climate does southern California maintain?", "id":"5705fc3a52bb89140068976a"},{"answers":[{"answer_start":59, "text":"infrequent rain"},

{"answer_start":59, "text":"infrequent rain"}, {"answer_start":59, "text":"infrequent rain"}], "question":"Other than many sunny days, what characteristic is typical for the climate in souther California?", "id":"5705fc3a52bb89140068976b"},

{"answers":[{"answer_start":243, "text":"60's"}, {"answer_start":243, "text":"60's"}, {"answer_start":243, "text":"60's"}], "question":"What is the low end of the temperature range in summer?", "id":"5705fc3a52bb89140068976c"},

{"answers":[{"answer_start":353, "text":"very rare"}, {"answer_start":353, "text":"very rare"}, {"answer_start":353, "text":"very rare"}], "question":"How frequent is snow in the Southwest of the state?", "id":"5705fc3a52bb89140068976d"},

{"answers":[{"answer_start":269, "text":"70"}, {"answer_start":269, "text":"70"}, {"answer_start":269, "text":"70"}], "question":"What is the high end of the temperature range in winter?", "id":"5705fc3a52bb89140068976e"}] }

Figure 3. Example from SQuAD: Passage (context) from a Wikipedia article, examples of questions about the passage, and potential answers identified in the text (and their location in the text)

• All: proportion of the candidate answers present in at least one of the answers for the correspondent passage;

• Top 10: proportion of the ten best-scored candidate answers (accordingly to TF-IDF) present in at least one of the answers for the correspondent passage;

• Last Position (LP): position of the last candidate answer that appears in at least one of the answers for the correspondent passage;

• BLEU-1 (B-1), BLEU-2 (B-2), BLEU-3 (B-3) and BLEU-4 (B-4);

• Rouge-L (R-L).

AQG commonly employs metrics such as BLEU (1, 2, 3, and 4) and ROUGE-L to compare the similarity of two sections of text using concepts like ngrams and longest common sub-sequences. These metrics were used in both Answer Selection and Question Generation steps. To perform the evaluation with each metric, all candidates are compared with all references that belong to the same passage.

The comparison of answer selection methods can be seen in Table 1.

Table 1
Comparison of Answer Selection Methods

	All	Top 10	LP	B-1	B-2	B-3	B-4	R-L
Terms	0.2595	**0.4179**	73.2636	0.1901	0.0013	0.0013	0.0013	0.0178
Named Entities	**0.3509**	0.3635	9.8737	**0.3829**	**0.1598**	0.0755	0.0379	**0.0615**
Noun Chunks	0.2255	0.2620	26.9229	0.3212	0.1538	0.0643	0.0233	0.0485
Clauses	0.0186	0.0196	**5.9473**	0.2383	0.1023	0.0597	**0.0402**	0.0607
Bigrams	0.1200	0.1722	96.3537	0.2765	0.1235	0.0010	0.0010	0.0387
Trigrams	0.0795	0.1039	102.3848	0.2850	0.1316	**0.0859**	0.0009	0.0498
T+NEs+NCs	0.2471	0.3969	97.3652	0.2527	0.0615	0.0224	0.0075	0.0276

In addition to these metrics, we determined the proportion of candidate answers in at least one reference answer for the correspondent passage (All) and the position of the last common answer among all candidates and references sets (LP).

The scores for the various metrics and answer selection methods are presented in Table 1. The best value for each metric is bold while the "Named Entities" method has the highest score for "All" - a proportion of candidate answers found in one or more correspondent passages.

The evaluation process was repeated again, except for the stop words from selected and reference answers, to test whether their presence had an effect on the scores. The results were similar in both BLEU-3 and PLAF (although not specifically high scorers). Despite being relatively close, "Noun Chunks" was the best-scored item in BBEU-4 while "Trigrams"' resulted in defeat.

Excluding stop words, the comparison of answer selection methods is shown in Table 2.

Table 2
Comparison of Answer Selection Methods (without stop words)

	All	Top 10	LP	B-1	B-2	B-3	B-4	R-L
Terms	0.2569	**0.3861**	51.7271	0.2083	0.0018	0.0018	0.0018	0.0242
Named Entities	**0.3493**	0.3549	8.6710	**0.3694**	**0.1655**	0.0572	0.0259	**0.0662**
Noun Chunks	0.2047	0.1998	17.1316	0.2899	0.1618	**0.0731**	**0.0315**	0.0556
Clauses	0.0128	0.0128	**3.1532**	0.2190	0.0983	0.0443	0.0255	0.0631
Bigrams	0.1207	0.1407	61.2015	0.2664	0.1282	0.0016	0.0016	0.0474
Trigrams	0.0636	0.0692	58.3816	0.2716	0.1367	0.0725	0.0016	0.0588
T+NEs+NCs	0.2436	0.3522	70.0875	0.2549	0.0716	0.0263	0.0098	0.0345

For both sets of tests, the methods that favored single words ("Terms" and "T+NEs+NCs") had the second and third highest scores in "All" while the "Top 10" score was the top one (not including "NEm" which was second best when not considering stopwords). In these metrics, we assume that there is a candidate answer to be present in the references as well as it is contained in one, giving single word advantage over other words.

"Clauses" may have been the more effective method for evaluating "LP" since it generates numerous candidates, but this approach reduces the likelihood of them being contained by a reference.

The metric that selects answer candidates with exactly three words ("Trigrams") has a higher score in BLEU-3, "Clauses", and TLEU-4. However, the scores vary slightly when stop words are removed from these two methods, as most trigram/clans tend to have chunks of nouns composed of stop word, while others do not.

The "Named Entities" method was found to be the most consistent in scoring the highest, given the values of both groups of tests. As a result, names were used in subsequent tests to select answers.

We found a GitHub repository that had transformers capable of both answer-aware and answeragnostic selection methods. By isolating the part responsible for answer selection, we could combine it with question generation in accordance with rules to achieve optimal results.

The comparison of transformers used in the answer selection methods is documented in Table 3.

Table 3
Comparison of Answer Selection Methods (transformers)

	All	Top 10	LP	B-1	B-2	B-3	B-4	R-L
QG	**0.4365**	**0.4355**	**4.2199**	**0.4467**	**0.2705**	**0.1474**	**0.0884**	**0.0886**
QG Preprend	0.1200	0.1185	4.5787	0.4378	0.2617	0.1427	0.0864	0.0862
QG-QA	**0.4365**	**0.4355**	**4.2199**	**0.4467**	**0.2705**	**0.1474**	**0.0884**	**0.0886**

QG and QL-QA are the answer-aware transformers that can handle Question Generation and Question Answering. They differ in their approach to identifying the solution within the context, with Qg Prepend presuming the response to the given context.

As depicted in the table, QG and QGS-QA both exhibit comparable outcomes for this task, with no distinction between them. When evaluating "All" and "Top 10", they score much higher than their predecessors, but with a smaller gap. In contrast, these results are more consistent across all metrics except stop words.

The use of highlights resulted in better outcomes, but the prepend transformer's handling of answer selection in the pipeline was more challenging to isolate than other methods. Although it did not perform as well as the other method, it still delivered better results in most metrics compared to the earlier tested methods; nevertheless, we decided to use this transformer in subsequent methods when selecting answers.

Automatic Evaluation of Question Generation Methods

We proceeded to compare question generation methods. For this, we used not only the answer-aware techniques that were already evaluated for selecting answers (QG, QA-QGA and QG Prepend) but also tested an answer–agnostic transformer (E2E). These tests were conducted independently.

The answer-aware transformer, which uses the prepend format and is designed for question generation, was also chosen. As we had previously discussed, it was more efficient to isolate parts of the answer selection process using QG Prepend.

The outcomes of BLEU and ROUGE are presented in Table 4. We were taken aback to find that QG Prepend scored better than the other two answer-aware transformers from the same repository (QG and QA-QF) for all metrics except B-4. We also observed positive results for E2E, which scored slightly lower in almost all but not B-5. At every metric, [20] (with answers selected with QGI and chosen by others) gave 03% score up to BOE

Table 4

Comparison of Question Generation Methods

	B-1	B-2	B-3	B-4	R-L
QG	0.5327	0.2409	0.1301	0.0774	0.2260
QA-QG	0.5403	0.2464	0.1347	0.0808	0.2291
QG Prepend	0.5459	0.2499	0.1348	0.0792	0.2336
E2E	0.5347	0.2434	0.1313	0.0804	0.2268
[Romero, 2021] (answers from QG Prepend)	**0.5576**	**0.2566**	**0.1389**	**0.0816**	**0.2377**
Rules (answers from NEs)	0.3853	0.1324	0.0638	0.0357	0.1549
Rules (answers from NCs)	0.3774	0.1223	0.0574	0.0311	0.1470

The rule-based approach, which is used for selecting named entities and noun chunks, had the worst values for this approach. This was not a surprise, given that transformers are considered state-of-the-art. However, the dataset used is also important as SQuAD is not comprehensive. Good questions can receive scores that do not reflect them well since there may be no similar questions in the same dataset. Transformers also have an advantage in being trained in constructing algorithms that train models with complex algorithms.

We demonstrated once again that the use of named entities rather than noun chunks was a viable option, as it was the version that scored the highest between the two rule-based approaches.

3. Conclutions

The focus of this work was on ways to address the challenge of creating multiple-choice questions automatically. The ultimate aim was to create a system that would utilize various approaches to generate multiple types of questions randomly, using different techniques. This pipeline was chosen to implement the system and it proved to be an ideal solution for integrating diverse approaches such as response selection, question generation, and distractor selection.

Some approaches have not been successful, particularly the rule-based approach. Although the use of rules was expected in comparison to "Transformers", it allowed for greater control over the question-generating process and provided a baseline for Transformer. However, as an out-of-the-box approach developed with limited assistance from certain linguistic analysis libraries (such as Aldrin Library), it was interesting to implement.

The transformer did not rate named objects or expressions on the Answer Selection task differently, and both options were satisfactory. However, in a more comprehensive evaluation, the methods with the answers chosen by the transformer were slightly better. With respect to distractors, we could create distors from both the source text and external sources.

There are other areas that could be improved in future work. This includes non-machine learning approaches to question generation, where we can improve the rules or explore methods that were not tested previously, such as SRL. Regarding distractors, we still need to be able to generate dispensers that vary in level of incorrectness, with some being more incorrect than others. Additionally, our approach did not include any means of validating and ranking the generated questions, which could enhance the quality of the provided questions to the user.

Our study was able to compare and combine various NLP techniques utilized for question generation, leading to the creation of an approach that, while still challenging, can be refined through experimentation. The results indicate that the integration of different approaches has resulted in successful outcomes for the AQG task by creating a pipeline that can perform each of the three sub-steps described above.

The development of additional systems could yield significant benefits in the future. By enhancing existing methods and considering more complex questions, such a system appears to have the potential to reduce time spent on creating tests and questionnaires and become an additional tool for education and training.

4. References

1. A. Belchikov, "Automated Testing Systems in Education: A Review," International Journal of Emerging Technologies in Learning (iJET), vol. 10, no. 1, pp. 4-12, 2015.
2. B. Tang and W. Lu, "Application of Neural Networks in Education Assessment," in 2019 International Conference on Education Technology and Social Science (ICETSS), Chengdu, China, 2019, pp. 251-254.
3. S. Liu, J. Zhang, and Y. Xie, "An Automated Knowledge Assessment System Based on Deep Learning," in 2020 IEEE 2nd Conference on Multimedia Information Processing and Retrieval (MIPR), Beijing, China, 2020, pp. 214-218.
4. J. Zhang, Y. Cui, and H. Wang, "An Automated Assessment System for Students' Learning Outcome Based on Deep Learning," in 2021 IEEE 13th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 2021, pp. 98-103.
5. M. A. M. Hashim and S. S. Sabirin, "A Neural Network-based Adaptive Assessment System for Mathematics Learning," in 2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS), Kuala Lumpur, Malaysia, 2019, pp. 1-6.
6. A. Pal, A. Saha, and S. Chakraborty, "An Efficient Automated Assessment System for Students using Neural Network," in 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2018, pp. 190-196.
7. N. Thangaraj and M. A. T. Ramalakshmi, "Neural Network based Assessment System for Evaluating Programming Skill of Students," International Journal of Computer Applications, vol. 174, no. 13, pp. 7-12, 2017.

8. S. P. R. Jha, "An Intelligent Automated Assessment System using Neural Networks," in 2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), Chennai, India, 2017, pp. 1-5.
9. R. Zafar, M. A. Majeed, and M. Yaqoob, "Automated Assessment of Student Knowledge in E-learning Systems using Neural Networks," in 2020 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Bangkok, Thailand, 2020, pp. 35-40.
10. X. Wang, J. Gao, and X. Zhang, "Development of an Automated Assessment System for Chemistry Experiments based on Deep Learning," in 2021 13th International Conference on Education Technology and Computers (ICETC), Osaka, Japan, 2021, pp. 58-62.
11. M. C. Martinez-Torres, R. A. Rodriguez-Diaz, and L. C. Castro-Beltran, "A Review of the Use of Neural Networks in Education," IEEE Latin America Transactions, vol. 17, no. 9, pp. 14111416, 2019.
12. Y. Liu and Y. Dong, "Automated Assessment System for Students' Learning Outcomes based on Deep Learning," in 2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT), Tartu, Estonia, 2020, pp. 195-199.
13. S. S. Sabirin and M. A. M. Hashim, "Neural Network-based Assessment System for Learning English as a Second Language," in 2018 IEEE 5th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Songkhla, Thailand, 2018, pp. 1-6.
14. A. H. A. H. A. Ghaffar, M. A. R. Abro, and H. A. M. Jamali, "Automated Assessment System for Students' Writing Skill using Artificial Neural Network," in 2021 IEEE 15th International Conference on Innovations in Information Technology (IIT), Abu Dhabi, United Arab Emirates, 2021, pp. 1-6.
15. M. A. Majeed, R. Zafar, and S. B. Qaisar, "An Automated System for Assessing Students' Knowledge in Physics using Neural Networks," in 2021 IEEE 18th International Conference on Smart Communities: Improving Quality of Life using ICT, Lahore, Pakistan, 2021, pp. 104-108.
16. X. Liu, Y. Chen, and Z. Gao, "Design and Implementation of a Knowledge Assessment System for Distance Education based on Neural Network," in 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, 2019, pp. 10-14.
17. K. Wu and Y. Zhang, "Automated Assessment of Learning Outcomes based on Deep Learning," in 2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT), Tartu, Estonia, 2020, pp. 5-9.
18. J. Han, J. Wu, and Y. He, "Development of a Knowledge Assessment System for Distance Education based on Deep Learning," in 2021 11th International Conference on Intelligent HumanMachine Systems and Cybernetics (IHMSC), Nanchang, China, 2021, pp. 154-158.
19. Сіткар Т. В. Реалізація інтелектуальної інформаційної системи тестування з відкритою формою тестового завдання / Т. В. Сіткар // Науковий часопис Над. пед. ун-ту ім. М. П. Драгоманова. Серія 5. Педагогічні науки: реалії та перспективи: зб. наук. пр. / за ред. В. П. Сергієнка.-К., 2011.-Вип. 28. — С. 231-237
20. Manuel Romero. T5 (base) fine-tuned on squad for qg via ap. https://huggingface.co/mrm8488/t5-base-finetuned-question-generation-ap, 2021.
21. Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. arXiv preprint arXiv:1704.04683, 2017
22. Ho-Lam Chung, Ying-Hong Chan, and Yao-Chung Fan. A bert-based distractor generation scheme with multi-tasking and negative answer training strategies. arXiv preprint arXiv:2010.05384, 2020.