International Journal of Modern Developments in Engineering and Science
Volume 2, Issue 3, March 2023
https://www.ijmdes.com | ISSN (Online): 2583-3138

Student Automatic Correction Using Natural Language Processing

A. Saranya[1*], M. Monisha[2], P. Nandhini[3], M. Rakshana[4], R. Tejashree[5]

[1]*Assistant Professor, Department of Computer Science and Engineering, Vivekanandha College of Engineering for Women, Tiruchengode, India*
[2,3,4,5]*Student, Department of Computer Science and Engineering, Vivekanandha College of Engineering for Women, Tiruchengode, India*

***Abstract*: In this paper, we present a system for evaluating the quality of a question paper automatically. The question paper is an important part of educational assessment. The quality of a question paper is critical to achieving the assessment's goal. Question papers are prepared by hand in many educational sectors. Prior analysis of a question paper may aid in identifying errors in the question paper and better achieving the assessment's goals. We will concentrate on higher education in the technical domain in this experiment. First, we conducted a student survey to identify the key factors influencing question paper quality. We identified three key factors: question relevance, question difficulty, and time constraint. Automatic grading necessitates the use of cutting-edge technology. The traditional evaluation system entails a specific evaluator of a specific subject manually evaluating the answers written by students using Natural Language Processing (NLP). Manually evaluating answer scripts for each student is a time-consuming process for an evaluator using Recurrent Neural Network (RNN). As a result, an automatic exam paper evaluation framework has been proposed to enable an automatic answer correction and grading system. This system can be used in any educational institution to reduce the time spent manually evaluating answer scripts. The proposed automatic exam correction framework for digital answers [AECF] was created using Python programming languages and Python packages such as NLTK. For the backend, a Python flask server was used, and the UI was enhanced with HTML and CSS.**

***Keywords*: Educational assessment, Natural Language Processing, Recurrent Neural Network, NLTK.**

1. Introduction

The goal of this article is to create a framework for estimating the quality of a question paper automatically. The most common type of assessment is question paper-based. The method in various sectors of educational assessment. A question paper is prepared for this purpose to assess how well a student can demonstrate their acquired knowledge and understanding. A question paper is a collection of questions. A question is made up of two parts: a short text that requests information from the responder and a maximum score that will be awarded to the responder based on the correctness of the answer. In today's world, there are numerous exam administration methods available, such as online exams, OMR sheet exams, and MCQ-type exams. Every day, various examinations are held all over the world. The most important aspect of any examination is the checking of the student's answer sheet. Typically, it is done manually by the teacher, making it a time-consuming task if the number of students is large. In such a case, automating the answer-checking process would undoubtedly be beneficial.

Automating the answer-checking process would not only relieve the exam checker, but it would also make the checking process far more transparent and fair, as there would be no chance of bias from the teacher's side. There are numerous online tools for checking multiple-choice questions nowadays, but there are very few tools for checking subjective answer-type examinations. The goal of this project is to use machine learning to check subjective answer-type examinations. This application can be used in a variety of educational settings to check subjective answer-type examinations.

Furthermore, as the application is improved, it can be expanded to conduct online subjective answer type examinations. When the application is launched, the user will be presented with two options: login as an administrator or as a student. After selecting one of the options, the user will be presented with a login window in which he/she will be prompted to enter his/her credentials. The administrator will have the option of uploading the question paper and viewing the students' responses. The student will be able to add the answer and see the marks assigned to them there.

2. Literature Survey

People's competition has increased significantly in today's world. With the world's population growing, competition among people can be seen everywhere, as everyone wants to live the life of their dreams. Everyone aspires to be better than their peers. Another major reason for increased competition is a scarcity of resources, particularly jobs if we limit our study to the professional world. This competition starts early in one's life, in schools and colleges. Exams in schools and colleges are used to determine who is better academically than others. Simply put, the student with the highest-grade point average is considered to be the most intelligent.

The automatic evaluation of question papers is a relatively unexplored territory. There have been a few attempts in the literature to assess the quality of computer-generated questions. Automatic question generation (AQG) is a related research area

*Corresponding author: saranya@vcew.ac.in in which many efforts have been made to develop systems for the automatic generation of questions from text. Kurdi et al. provide an overview of the literature on automatic question generation for educational purposes (2019). Automatic multiple-choice question generation is a subproblem of AQG; Rao and Saha provide a review of the literature on automatic MCQ generation (2020). The researchers proposed several approaches and metrics for assessing the quality of system-generated MCQs.

Chali and Hasan (2015) concentrated on determining the questions' syntactic correctness. They proposed a method for calculating the syntactic similarity of each question to its associated content information. Araki et al. (2016) graded the questions on grammatical accuracy and distractor quality. Narendra, Agarwal, and Shah (2013) evaluated the quality of the system generated questions based on their informativeness and relevance. Zhang and VanLehn (2016) evaluated the quality of the system-generated questions using several parameters such as relevance, fluency, ambiguity, pedagogy, and depth. Susanti, Tokunaga, Nishikawa, and Obari (2017) proposed item analysis as a method for evaluating system generated questions.

Item analysis employed two parameters, difficulty index and discrimination index, to assess the quality of MCQs used in a test. Pandarova et al. (2019) sought to estimate the difficulty scoring of grammar exercise items for use in dynamic difficulty adaptation in a machine-learning-based language tutoring system. Luger and Bowles (2013) and Luger (2016) proposed several methods for assessing the quality and difficulty of multiple-choice questions. They proposed a method for automatically determining the difficulty and discriminating power of multiple-choice questions. Their approach to measuring question difficulty is based on models of how well-performing students will perform and compares them to their lower-performing peers.

3. Existing System

In the existing system, a normal examination traditional correction pattern is done. In the past ten years, assessment using new technology has solidified its position inside the university system. This essay provides an overview of the two years of digital examination implementation at Chennai, from the views of teachers, students, and administrators, these experiences are presented.

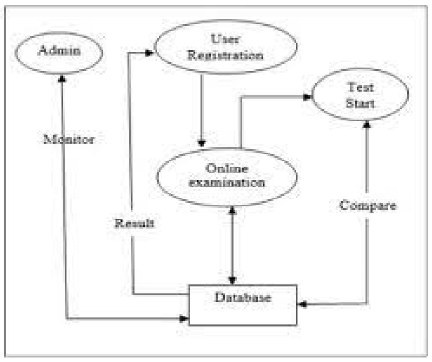

Fig. 1. Architecture of existing system

From the standpoint of the teachers, the experience has been quite beneficial. Less time was allotted for assessing written exams, the grades are thought to be fairer, and the time saved may be used to improve other course components.

4. Methodology

The proposed system for assessing the quality of a human-generated question paper. The first step is to identify the most important factors.

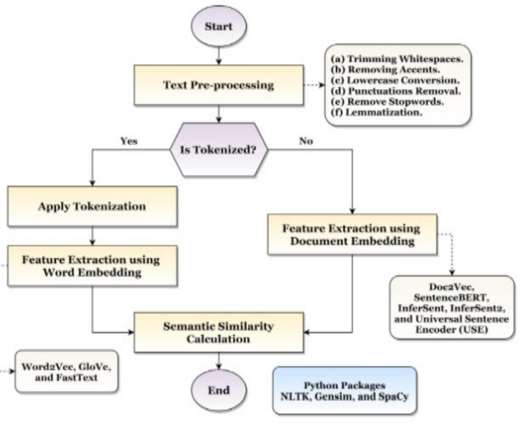
Start
Text Pre-processing
(a) Trimming Whitespaces.
(b) Removing Accents.
(c) Lowercase Conversion.
(d) Punctuations Removal.
(e) Remove Stopwords.
(f) Lemmatization.
Yes
Is Tokenized?
No
Apply Tokenization
Feature Extraction using Document Embedding
Feature Extraction using Word Embedding
Doc2Vec, SentenceBERT, InferSent, InferSent2, and Universal Sentence Encoder (USE)
Semantic Similarity Calculation
Word2Vec, GloVe, and FastText
End
Python Packages NLTK, Gensim, and SpaCy

Fig. 2. Proposed system architecture

The students' opinions are extremely important in determining the quality factors. So, in order to better understand the quality factors, we conducted a survey. The pycharm system was used to run the portion of the application that uses Machine Learning to analyze the student's answer sheet. This Notebook is widely used for performing and carrying out projects and experiments in the field of data science and data visualization. It is a web-based open-source tool. This notebook is used by the majority of machine learning applications available.

- Train question and keyword
- Online Exam
- Automatic Correction
- RNN classification
- Marks allotment

1) Train questions and keyword

The first step is to collect the data set, which consists of answers to the questions on the question paper. The admin has the option in this system to upload the question paper as well as see which students have submitted the answer sheet. When the student successfully logs into the system, he or she will be able to view and download the question paper. When the user opens the application's main window, he or she is presented with two options: Staff and Student. The user can proceed by selecting one of the options.

2) Online examination

The first layer is in charge of accepting the student's responses to various questions. The second layer is in charge of combining each student's response with the corresponding reference response. A short answer's reference answer can be keywords or the actual answer. The third layer is the Auto Correction layer, which examines the question type and determines whether it is a single step or a series of steps. The instructor should anonymously distribute each student's answer (along with its correction). The fourth layer is in charge of storing the results in a storage medium (such as a database or cloud storage) and displaying the final report to the user. It does not produce new headers.

3) Automatic correction

The process of determining the semantic similarity percentage between two texts is known as semantic text matching [1]. They go through (1) text pre-processing, (2) tokenization (the text can be used whole or tokenized into words), (3) feature extraction, and (4) semantic similarity scoring. Figure 1 summarizes the process, which is further discussed in the following subsections.

The pre-processing stage consists of the following steps: (1) trimming (removing) whitespaces from the left and right of the text, such as spaces, tabs, and new line feeds, (2) removing accents from the text, (3) converting the entire text to lowercase, (4) removing punctuation wisely, (5) removing stopwords, and (6) applying lemmatization.

Tokenization is the process of converting (breaking) text into components, pieces (words and sub-words), punctuation, and other elements [7]. The breaking is done with a delimiter, which is commonly "space." The tokenization algorithm should check each tokenized element and respond to the question "Should this element be tokenized or not?" For example, "the United States of America" should not be tokenized, whereas "the Earth has" should be tokenized. Abbreviations, emails, and website links should also be added to the list of exceptions. If we want to work with the entire text or paragraph [8,] we can skip this phase.

The TF-IDF is a refinement of the TF in that it downscales the weights for words that appear in a large number of documents in the corpus and are thus less informative compared to those that appear in a smaller portion of the corpus and are more informative. The term IDF is a logarithmically scaled inverse fraction of the documents containing the term, whereas TF-IDF is a combination of the count vectorizer, TF, and IDF factors. In other words, the TF-IDF indicates how significant a word is in the document

4) RNN classification

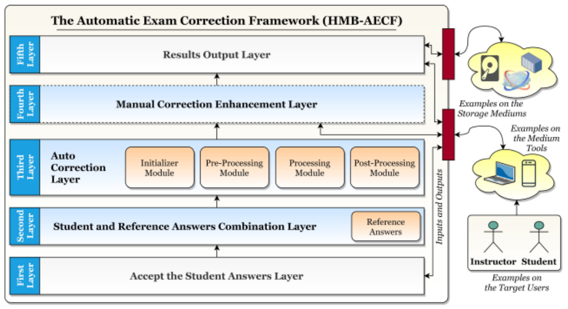
The Automatic Exam Correction Framework (HMB-AECF)
Fifth Layer
Results Output Layer
Fourth Layer
Manual Correction Enhancement Layer
Third Layer
Auto Correction Layer
Initializer Module
Pre-Processing Module
Processing Module
Post-Processing Module
Second Layer
Student and Reference Answers Combination Layer
Reference Answers
First Layer
Accept the Student Answers Layer
Inputs and Outputs
Examples on the Storage Mediums
Examples on the Medium Tools
Instructor
Student
Examples on the Target Users

Fig. 3. The RNN classified text matching layers

The auto-correction layer is made up of four modules: (1) the initializer, (2) the pre-processing, (3) the processing, and (4) the post-processing. The initializer module examines the question type (from the first layer) to determine whether the question is a single step or a series of steps. It encapsulates the single-step question into an array of a single element to generalize the two cases using loops or queues.

1. function PREPROCESS EXPRESSION (expression) \\ The function accepts the expression and returns the pre-processed expression.
2. trimmed Expression← Remove Whitespaces (expression) \\Remove the whitespaces from the expression.
3. norm Expression, lookup Table ← Normalize (trimmed Expression) \\Normalize the expression and get the normalized expression and the lookup table.
4. return norm Expression, lookup Table \\ Return the pre-processed expression with the lookup table.
5. function PROCESSEXPRESSION (expression, lookup Table) \\The function accepts the expression and the merged lookup table. It returns the processed expression tree.
6. replaced ← Replace(expression) \\The square brackets are replaced with parentheses in the expression.
7. prefix ← Infix2Prefix(replaced) \\The prefix notation is extracted.
8. expression Tree ← Build Expression Tree(prefix) \\The expression tree is built.
9. expanded Tree ← Expand Expression Tree (expression Tree) \\ The expression tree is expanded.
10. refined ← Refine Tree (expanded Tree) \\The expanded tree is refined.
11. restored ← Restore Expression (refined, lookup Table) \\ The refined expression is restored using the lookup table.
12. cleaned ← Remove Curly Braces(restored) \\ Remove the curly braces from restored expression.
13. result ← Stringify (cleaned) \\Stringify the cleaned expression.
14. return result \\ Return the processed and expanded expression tree.
15. function POSTPROCESSEXPRESSION (first, second) \\The function accepts the two processed expressions and returns the similarity score.
16. similarity ← Calculate Similarity(first, second) \\Calculate the similarity score between the two expressions.
17. return similarity \\Return the similarity score.
18. function HMB-MMS-EMA (first Expression, second Expression) \\ The function accepts the two expressions and returns the similarity between them.
19. first Preprocessed, first Lookup Table ← Preprocess Expression (first Expression) \\Pre-process the first expression.
20. second Preprocessed, second Lookup Table ← Preprocess Expression (second Expression) \\Pre-process the second expression
21. lookup Table ← Merge Lookup Tables (first Lookup Table, second Lookup Table) \\Merge the two lookup tables.
22. first Processed ← Process Expression (first Preprocessed, lookup Table) \\Process the first pre-processed expression.

```
23. second Processed ← Process Expression (second Preprocessed, lookup Table) \\Process the second pre-processed expression.
24. similarity ← Postprocess Expression (first Processed, second Processed) \\Post-process the two processed expressions.
25. return similarity
```

5) Marks allotments

This layer is in charge of saving the results to a storage medium (such as a database or cloud storage) and displaying the final report to the user. It does not produce new headers. Students had written the issues using their own words. These statements were manually normalized during the analysis. We discovered that the majority of participants raised three concerns. Another important consideration in timed examinations is the length of the question paper or response time. The length of the responses to the entire question paper should be answerable within the time limit. If one question is too long, another should be added to balance it out so that the overall question paper is manageable.

5. Result and Discussion

Aspects in order to gain a thorough understanding of the accuracy of our proposed method. The first aspect is based on quality. To assess the accuracy of our system, the Automatic Answer Checker was evaluated on the basis of quality. The second aspect evaluated was performance, which was done to gain a through understanding of the comparison of automated answer checking method with traditional answer checking method. A survey was conducted solely to assess the quality of the automatic answer checker system. This survey was administered to a group of university students as well as faculty members from various departments at the university.

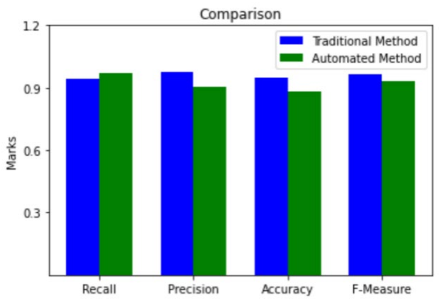

Fig. 4. Comparison between traditional and automated methods

On the basis of evaluation measures, the figure depicts the difference between the automated approach and the traditional approach. The Recall, also known as the True Positive Rate, was calculated to be 0.9712 for the automated approach and 0.9443 for the traditional approach.

Precision, also known as Positive Predictive Value, was found to be 0.9062 in the automated approach and 0.9771 in the traditional method.

Accuracy, also known as True Results, was found to be 0.8800 in the automated approach and 0. 9500 in the traditional method.

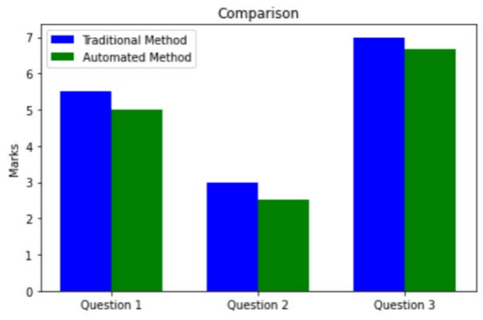

Fig. 5. Marks allotment on questions equal to traditional method

The above bar graph clearly shows the difference between the marks awarded by the system and the marks awarded to the student using the traditional approach. As can be seen, the difference between the marks given by the system and the marks given by the traditional method is marginal, indicating that our system was designed with high accuracy and precision. This further demonstrates the dependability of our designed project in correctly predicting the student's grades.

Because our automatic answer checker is related to their domain, selecting such a sample was obvious. The survey sample was almost evenly split between male and female participants. When all of the survey results were analysed, we discovered that 85% of the survey participants strongly agreed and 15% agreed that our designed system provided precise results when they used it. 77% of survey respondents strongly agreed, and 23% agreed that the efficiency of our designed system was fast enough to complete all of their tasks.

97% of survey participants strongly agreed, and 3% agreed that the designed system was quite user friendly and easy to use, even if they had never used anything similar before. The designed system fulfilled all of the user's operations, according to 95% of survey participants who strongly agreed and 5% who agreed. To summarise, every person who participated in the survey was extremely satisfied with the designed system. As a result, our designed system performed admirably in the quality evaluation test.

6. Conclusion

The project report titled automatic answer checker is now in its final stages. The application was created with every possible error in mind, so the system is quite efficient and reliable. Because the application has the unique property of being robust in nature, there are numerous ways to implement improvements in the application in the near future. The application would be approved and authenticated shortly before being implemented. The purpose of this paper was to present a novel technique for evaluating the quality of a question paper using a computer platform. A student survey was used to identify the key factors that influence the quality of a question paper. Various techniques were used to implement individual modules for automatic assessment of those factors. A set of real question papers was used to test the implemented modules. According to the evaluation results, the proposed technique is effective in estimating question relevance, question paper difficulty, and response time estimation.

Future work would consist of developing an assessment algorithm whose purpose would be to detect all syntax errors in our keywords, and then investigating it for high performance and equality in addressing them.

References

[1] R. Mihalcea, C. Corley, C. Strapparava et al., "Corpus-based and knowledge-based measures of text semantic similarity," in AAAI, vol. 6, no. 2006, 2006, pp. 775–780.

[2] L. Zhiqiang, S. Werimin, and Y. Zhenhua, "Measuring semantic similarity between words using wikipedia," in 2009 International Conference on Web Information Systems and Mining. IEEE, 2009, pp. 251–255.

[3] E. T. Al-Shammari, "Lemmatizing, stemming, and query expansion method and system," Jun. 25 2013, US Patent 8,473,279.

[4] V. Balakrishnan and E. Lloyd-Yemoh, "Stemming and lemmatization: a comparison of retrieval performances," 2014.

[5] N. Habash, O. Rambow, and R. Roth, "Madatokan: A toolkit for arabic tokenization, discretization, morphological disambiguation, pos tagging, stemming and lemmatization," in Proceedings of the 2nd international conference on Arabic language resources and tools (MEDAR), Cairo, Egypt, vol. 41, 2009, p. 62.

[6] I. Boban, A. Doko, and S. Gotovac, "Sentence retrieval using stemming and lemmatization with different length of the queries," Adv. Sci. Technol. Eng. Syst. J, vol. 5, pp. 349–354, 2020

[7] R. M. Kaplan, "Method and apparatus for tokenizing text," Feb. 24 1998, US Patent 5,721,939.

[8] P. Mcnamee and J. Mayfield, "Character n-gram tokenization for European language text retrieval," Information retrieval, vol. 7, no. 1-2, pp. 73–97, 2004.

[9] G. Carenini, R. T. Ng, and E. Zwart, "Extracting knowledge from evaluative text," in Proceedings of the 3rd international conference on Knowledge capture, 2005, pp. 11–18.

[10] N. Azam and J. Yao, "Comparison of term frequency and document frequency-based feature selection metrics in text categorization," Expert Systems with Applications, vol. 39, no. 5, pp. 4760–4768, 2012.

[11] A. M. Dai, C. Olah, and Q. V. Le, "Document embedding with paragraph vectors," 2015.

[12] M. Mohd, R. Jan, and M. Shah, "Text document summarization using word embedding," Expert Systems with Applications, vol. 143, p. 112958, 2020.

[13] A. Kulkarni and A. Shivananda, "Converting text to features," in Natural Language Processing Recipes. Springer, 2019, pp. 67–96.

[14] H. Christian, M. P. Agus, and D. Suhartono, "Single document automatic text summarization using term frequency-inverse document frequency (tfidf)," ComTech: Computer, Mathematics and Engineering Applications, vol. 7, no. 4, pp. 285–294, 2016.