e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:03/March-2025 Impact Factor- 8.187 www.irjmets.com

LEVERAGING NLP AND TRANSFORMER MODELS FOR EFFICIENT AND ACCURATE DESCRIPTIVE ANSWER EVALUATION

Kapil Damodare[*1], Prajyot Rasekar[*2], Sarthak Borude[*3], Nayan Khairnar[*4]

[*1,2,3,4]IT Engineering, PDEA College Of Engineering, Manjari(BK), Pune, India.

DOI: https://www.doi.org/10.56726/IRJMETS68476

ABSTRACT

This project aims to implement a real-time evaluation system that assesses descriptive answers using advanced natural language processing (NLP) and machine learning techniques. Manual grading is the standard procedure for evaluative response systems; more so in the case of descriptive and open-ended answers that are very laborious, inconsistent, and non-scalable. Therefore, this project proposes an automated system that analyzes the content, grammar, coherence, and relevance of the given descriptive answers among other parameters with high accuracy. First, the system uses NLP models to analyze the syntactic and semantic structures of students' responses. This content will be processed and compared with pre-set evaluation and sample answer strategies, and a score is provided through the model during interaction. Due to the popularity of such architectures in natural language understanding, transformer-based models will be also applied, especially those such as BERT or GPT. Other techniques also include semantic similarity, sentiment analysis, and keyword matching, which will allow understanding of the varying aspects of students' answers while upholding the quality and fairness of the assessment. The envisaged solution will provide a consistent and efficient evaluation mechanism, which will ultimately be a multiplier of benefits for educational institutions while lessening the burden on teachers and simultaneously allowing for instant response to learners. The result of this particular endeavor will be an accurate evaluation mechanism suitable for educational systems implementing descriptive answer scoring using automated means and high scalability.

Keywords: Descriptive Answer Evaluation, Cosine Similarity, Machine Learning, Natural Language Processing, Word2vec, Stop Word Removal, Stemmer.

I. INTRODUCTION

The open-ended nature of subjective questions and answers allows for the evaluation of a student's performance and ability in any aspect. Unsurprisingly, the responses were not limited to any boundary, and students were at liberty to compose them based on their perception and comprehension of the concept. However, there are a number of other important differences between subjective answers and those objectives. Starters surpass elementary starters in size. Additionally, they require more time to complete. In addition, they provide much more context, which requires a lot of attention and subjectivity from the person grading them. This raises difficulties, particularly because natural languages are full of syntax notations. Doing so requires considerable data processing, cleanup, and tokenization. Subsequently, several approaches can be used with textual data, such as document similarity, latent semantic analysis, graph concepts, and ontologies. The overall grade can be determined according to similarity, presence of certain keywords, the way the text is organized, and the language [1], [2]. Numerous attempts have been proposed in earlier works to provide a solution to this issue [3]–[5]; however, there is still some scope for enhancements that are discussed in this paper. Both students and teachers find and hold subjective examinations viciously intricate and intimidating, mainly due to one major aspect, context. For a subjective answer, the answer scorer, the checker, must actively engage in with the answer attempting to score every sentence of the answer, which includes factors such as sickness, tiredness, or even subjectivity of the checker, as these aspects greatly affect the end result. Hence, it is a better time and resources economy to allow a system to perform this tiring, yet a slightly important assessment for judging the subjective answers. In contrast, evaluating objective answers using machines is simple and practical.

It is possible to provide the computer with a set of questions together with keywords that enable quick mapping of the students' answers to the expected responses. However, this becomes much more difficult with subjective answers. They differ in size and pose myriad vocabulary. In addition, people tend to use different words to mean the same thing as well as different forms of writing short words, which further complicates the

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:03/March-2025 Impact Factor- 8.187 www.irjmets.com

process. This level of concern has led to widespread research activities aimed at the evaluation of answers, especially subjective ones, and more specifically, to research inhibition, such as analyzing logically interacting phrases. There are many types of differences that may be assessed, including distances between individual two-dimensional shapes that may contain images that are separated by some underlying text, assessing the presence of key elements of the narratives, identifying the presence of keywords in provided responses, etc. But there are situations, such as overstated sentence and word relevance at the same time limit [6], problems with parameters of the model [7], expensive optimization of the approach [8], and issues with the quality of the resources [5] that persist. In this study, we designed a subjective answer evaluation algorithm using the evaluation methods of machinelearning and natural language processing techniques. The approach we used consists of the following components: tokenization and lemmatization techniques to process texts, TF-IDF and Bag of Words and word2vec text representation models, techniques to measure distances, such as cosine similarity or word mover distance, and classification methods, such as multinomial Naive Bayes. We employed different metrics, such as F1-score, Accuracy, and Recall, to compare the performance of the models. Additionally, we present some techniques that have been used for the evaluation of subjectively generated content or even the evaluation of text similarity in more general contexts. Nevertheless, in the case of subjective answers, the following are some minimum constraints imposed on the system: Synonyms are likely to exist in previous research. • Previous studies tended to have very large bounds. • Previous research has shown that the sentences within it are not ordered in any proper sequence. This manuscript introduces an innovative and refined technique for the automatic evaluation of descriptive question answers using machine learning, natural language processing, and deep learning paradigms. It uses 2 step approach is used to solve this problem. First, the answers are evaluated with respect to the solution and provided keywords using various similarity-based techniques, such as the word mover's distance. These are then trained to detach models where answers are evaluated but not solutions and keywords. For example a question that is subjective for instance "Give me the capital city of Pakistan and the reason why is it famous", would have the answer bearing in mind that " Pakistan's capital city is Islamabad and its famous for mountainous views". Before assessing the answer provided by the student to the posed question, two other components, the question and the answer, are also introduced into the system (in this case, useful keywords whose place will be taken difference by the context will be islambad and mountain scenery), and the system compares students' answers in terms of both similarity (contextually in this case, the modal answer and the students' answer) and the presence or absence of certain keywords. For example, a student can answer and say "Karachi is the capital city of Pakistan, as it is well-known for the mountains". In this case, such an answer could be given to approximately 50% of the total marks. In another case, "Islamabad and mountains surroundings" could be awarded around 30% of the total marks because the two most important keywords are there, but even the context is absent. And 'Islamabad is the capital city and it is well known that mountain scenery' could earn the full 100% for that answer because it wisely combines the contextual similarities and relevant keywords to the correct answer.

II. LITERATURE SURVEY

Previous research on utilizing Natural Language Processing (NLP) to evaluate students' open-ended responses has primarily concentrated on analyzing keywords rather than grammatical structures. Various academics and investigators have contributed to this field, each focusing on distinct aspects and facing different constraints. A study conducted by Ms. Shweta M. Patil and Prof. Ms. Sonal Patil examined the descriptive assessment of answers using NLP. This paper investigates the application of word processing technology in evaluating student learning outcomes. CAA systems have always concentrated on objective-type questions but are incapable of any higher-order questioning in a descriptive answer. This paper presents the case of yet another system designed to assess answers containing a succession of sentences, going beyond sentence-level analysis. This system applies statistical matching, information retrieval, and a complete NLP. It targets at achieving correct scoring, provision of feedback and room for improvement of The limitations of current CAA systems, especially on the higher-order levels of Bloom's taxonomy, where the assessment of student performance is concerned, have been addressed by this system. The team of Prof. S.P. Raut and Prof. S.D. Chaudhari is responsible for this study, which aims to automatically analyze answer scripts through the use of natural language processing. A discussion of four similarity measures, Co-sine, Jaccard, Bigram, and Synonym, is presented in this paper. The automated evaluation of answer scripts has been shown to extend the practicality of evaluation in beneficial

ways, and in some instances, is able to produce marks comparable to those awarded by human professors. This presentation also discusses the reasons for the automated analysis,the areas covered, and the structure of the paper itself. In addition, it presents a cross-sectional view of the existing techniques pertaining to subjective answer analysis beginning from statistical methods, information extraction, and a full-blown NLP approach. This technical background section describes NLP techniques, such as tokenization, stopword removal, POS tagging, lemmatization, stemming, and case folding. It further presents embedding techniques such as the Bag of Words and Word2Vec. The paper ends by discussing the merits of using these automated evaluation systems and their application possibilities in the educational field to enhance efficiency, cut costs, and reduce the chances of human errors[3]. Together with, which aims to reduce the time and labor required to assess any written test manually, machine learning is furnished to propose an automated answer evaluation system. The integrated system uses OCR, backpropagation algorithm, ReLU, ANN, CNN, RNN, and CRNN. It seeks to access theory-based answers by locating specific words and allocating them full marks irrespective of the completeness of the answer. In the said methodology First, the answer sheet is scanned, keywords are extracted using OCR, and marks are awarded based on the matching of those keywordsthe and length of the answer. The results of the system tend to be manually evaluated and demonstrate a positive effect on the time and quality of performance[2]. Mr. N. Dave, Mr. H. Mistry, and Mr. J. P. Verma, all MCA students, focused on the study of subjective answer checking with the occurrence matrix. The Research paper in This study elaborates on different techniques used for text comparison and similarity detection, such as latent semantic analysis (LSA), TF-IDF, cosine similarity,Euclidean distance, and Manhattan distance [4]. It also gives the features of an occurrence matrix in terms of containing word frequency and gives the definition of various preprocessing approaches such as data extraction, stopword removal, and stemming.The research presents a block diagram and claims the spell check can be done through the use for Jortho, a java API. It creates a framework for computing marks for answers,checking for spelling errors, computing a similarity index, and computing the overall marks by considering question and answer grading. This paper provides the text comparison process, which begins with the choice of the source file, prepares it, builds occurrence matrices of the source file and the student's answer sheet, and measures their difference using the Euclidean distance,distance, Manhattan distance, and the method invented by the authors themselves. A case study involving an experiment is shown, where samples, content prepared, content with occurrence matrices, and calculations of primary similarity percentage using various distance-measuring characteristics are demonstrated. Prof U. Hasanah who heads the Information and communications Technology Management study program along with Prof. A.E Permanasari, Prof. S.S Kusumawardani and Prof F.S Pribadi conducted a case study concerning numerous research carried out during in the Information Extraction era (IE). This study analyzed computer-assisted assessment systems for short-answer essays in education. The authors evaluated several models of the assessment, orienting more towards the techniques of uch extraction; however, these techniques concern only the matching of the student answer with the teacher answer, such as in parse tree matching, regular expression matching, any Boolean phrase matching, syntactic pattern matching, syntactic semantic pattern matching, semantic word matching, and LRS representation matching. This paper showcases various works of the domain, outlines the common datasets, and the text preparation steps that were taken before analysis. It also ranks the grading models in linear order with respect to the raters of humans. The most commonly used evaluation results, also in the grade-point index order, are accuracy agreement, kappa, and Pearson correlation. All in all, the paper covers the merits and demerits of automated grading system for short-answer questions and presents the scope of work that can be carried out in this field in the years to come. In fact, information extraction techniques exhibit very good performance in terms of both precision and information retrieval. The case study on Bert Model was conducted by Prof R. Devika, Prof. S. Vairavasundaram, and Prof. C. S. J. Mahenthar, Prof. V. Varadarajan and Kotecha proposed a model called Semkey-BERT, which utilizes a BERT-based sentence transformer to extract keypoints from Twitter data. It is well known that keyphrase extraction from Twitter data presents significant technical challenges. Therefore,the intention of the model is to seek improvements by using deep learning techniques and automatic feature extraction. In this model, BERT embedding and three different sentence transformer models were employed to fetch key phrases from tweets, and the key phrases were ranked using a rank aggregation technique. It has been demexperimentally demonstrated that the Semkey-BERT model outperformed the previous models with success rates of 86[6].

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:03/March-2025 Impact Factor- 8.187 www.irjmets.com

III. METHODOLOGY

The purpose of this research work is to analyze the descriptive answer script in an automatic cell and give marks to the concerned question respectively. In order to achieve this, we take answer script as input. Python programming language is used here for implementing every algorithm.

Then NLP is used to extract text from the answer script and process the data. Various similarity measure has been calculated that is used as the parameter for assigning marks.

3.1 Text Extraction

The captured image from the answer script has been used as input for text extraction. For extracting text from the image, a python class pytesseract has been used. Before extracting text, the noise from the image is removed to increase the extraction accuracy. Pytesseract is a class based OCR and has Unicode (UTF-8) support, and can recognize more than 100 languages. The result of pytesseract is shown in Fig. 1. and Fig.2. The extracted text has been used for further processing and computes various similarity measures.

3.2 Summary Generation

The process commences with the conversion of printed text into a digital format from which the extraction of text takes place. Furthermore, natural language processing is implemented to prepare an abstract of the lengthy text. With the help of summary generation, it will ease the task of processing texts by avoiding less important sentence from long text documents. There are many such approaches which are used for the auto summary writing. Emerging as a new trend, automatic text summarization —by selecting certain words or phrases from a larger piece of written work, is more effective. Here the average frequent words have been selected as keywords where most frequently occurring and least occurring words are not considered. Next, the importance of each sentence within the long text is determined in accordance with the number of keywords appearing in the sentence, which is raised to the second power and divided by the specified area. Threshold distance between two relevant terms in a sentence is often referred to as window size. Finally, the resultant weighted sentences are arranged according to their values in descending order and the first n sentences are selected as the summary for the lengthy text. Pseudocode representation of the text summarization algorithm might look like this. Summarization algorithm first receives text as an input. Secondly, it breaks the text into sentences or words. Thirdly, it prepares the word lists and removes duplicates. The forth step consists of the determination of the amount of each word appearing in the text. A formula for calculating the proportion that a single word occupies in relation to the total number of words is given. Fifthly, the stop words with the highest and lowest word percentage are excluded, and the average frequent words are chosen as keywords. Sixthly, consideration of the extent of the keywords to aid in determining window size for every sentence is explored. The weight of every sentence is determined by dividing the square of the number of keywords present in a sentence by the window size. Lastly, the explanation section weight value is arranged in decreasing order and the first n sentences are taken for the summary. Also, Another approach was implemented which relies on the bag-of-words omitting any regard to the keywords. Calculating precision, recall and f-measure has been helpful in ordering the possible techniques for the effective techniques of generating automatic summaries. The precision explains to what extent the summary provided by the system(summarization by machine) true to the fact.

Number of overlapping Sentence

Precision(P) = Number of overlapping Sentence/ Number of sentence in system summary

The recall explains to what extent the system summary is recovering the reference summary (which was generated by human).

Recall(R) = Number of overlapping Sentence /Number of sentence in reference summary

F-measure is the f-score on the correlation of the two measures that are precision and recall. The f-score includes precision scores and recall rates.

F-score = 2.P.R/P+R

In this instance, the evaluation metric based on keyword summarization presents a higher score than nonsensical text without keyword summarization. After that, the resulting summary is assessed against the correct answer in order to calculate several similarity indices. Summary generation methods and results have been elaborated further in the results and discussions section.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:03/March-2025 Impact Factor- 8.187 www.irjmets.com

3.3 Text Preprocessing

The output summary does have some redundant words that can be ignored to aid the next processing task on the text. The term used before is associated with applying certain techniques that make it possible for computers to understand application data. For instance, Natural Language Processing (NLP) comes in handy when performing text preprocessing due to its efficiency. Examples of text processing include: tokenize, remove StopWords, lemmatize, remove redundant words, etc.

For performing this preprocessing, NLTK is among the best toolkits available today for developing Python programs to manipulate data in the human language. It has the extremely helpful text processing functionality incorporated in it that works with fewer keystrokes. One of the built-in packages provided by NLTK called word_tokenize divides a given string and saves every individual component in a separate list. One of the most significant steps in text processing is to eliminate the words that do not contribute any significance. Such frequently used words that fail to add meaning to the context of the sentence are offered within a corpus of StopWords as per NLTK. The StopWord corpus has served the purpose of excluding irrelevant content. The next step in the process of text preprocessing is lemmatization of words. Many languages show variance in the form of a particular word. For instance, the word walk may exist as walking, walked and walks. Lemmatization is the method that tries fusing the word into one base form otherwise called the lemma. This would help in shortening the list of words thus saving on processing. To lemmatize every individual word, NLTK has a pre-defined function called WordNetLemmatizer that lemmatizes all the words into their respective lemmas. Where this data is to be used for performing any sort of application, the data has to be organized in a structure. One such structure is the bigram also called digram which refers to a pair of elements that are adjacent in a sequence of. The bigram frequency distribution is a widely used statistic for the purposes of assessing the degree of structural similarity between two or more pieces of text. In order to create bigram, a bigram function of NLTK is employed which gives a bigram of all the words together. Here also the frequency of occurrence for each word is counted and information is kept in one dictionary, where the word is a key and the number of times it occurs is the value in the dictionary.Then the frequency word dictionary and the bigram are used for the purpose of evaluating different types of similarity.

3.4 Means of Measuring

Similarity Often, it is necessary to assess whether two sentences differ or, in fact, are similar in content. Similarity measures determines the similarity or dissimilarity of two sentences taking into account a number of parameters. A number of similarity measure techniques are available to carry out such analysis. In this study, cosine similarity, Jaccard similarity, bigram similarity, synonym similarity are carried out. Cosine similarity is one such interesting similarity measure technique which compares two documents looking at the angle between them and measuring how similar or different are they. Cosine-similarity (A, B) = ---------- (4) ||A||.||B|| (A, B) Where A and B are the two word vector and each of the vector's component is a word frequency or a tfidf value. In this case, the cosine similarity index is computed based on the students answer to the question and the true answer provided. The Evaluators who have to assess similarity based on the cosine similarity indicies have very high expectations in regard to the outcome. In the course of this experiment, the tool for calculating the cosine similarity was developed in python. A pseudocode illustration of the building algorithm is provided below. Algorithm of Cosine Similarity 1. Take dictionary of word and frequency as input. 2. Create two word vector where one for student answer and another for true answer. Length of each vector should be the length of total word list. 3. Calculate dot product of two vector 4. Compute norm of first vector 5. Compute norm of second vector 6. Multiply first and second norm 7. Divide dot product result by multiplication result which will define cosine similarity. Another similarity measure technique is Jaccard Similarity measure which is used to find out similarity using intersection and union of two word lists. |A∩B| Jaccard Similarity (A,B) = ---------- (5) |A∪B| Where A and B are two word lists. Jaccard similarity is calculated by taking the word list intersection and place it over their union as a fraction. The intersection measures the commonality degree of two different word lists, while the union comprises all different words present in the two lists combined.

3.5 Marks Assessment

This research focuses on the incalculable determination of scores after conducting an evaluation. This becomes the last phase of the experiment, and its precision will improve the influence of the study. Here a weight value is

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:03/March-2025 Impact Factor- 8.187 www.irjmets.com

assigned to each criterion with respect to its significance. In order to obtain better results in weight value assignment accuracy, a surrogate study on approximate 50 samples has been done. Permitted and used is the average weight value estimation from the survey.

Marks = summation of Pk*Wk

Where Pk is the kth parameter and Wk. is the weight value of kth parameter. After assigning the weight value to each parameter, one simply multiplies the weight value with the parameter value. Finally all these multiplications values are added up to give the overall marks of that answer script. In order to test our experiment, it was done an evaluation in a manual way for thirty sample descriptive questions and the student answer has been considered for that question. This experiment takes into account three types of question in terms of marks. These are 5 marks question (M5), 10 marks question (M10) and 15 marks (M15) question It has been noted majority of the times, our proposed technique has obtained score marks very close to the manual scoring.

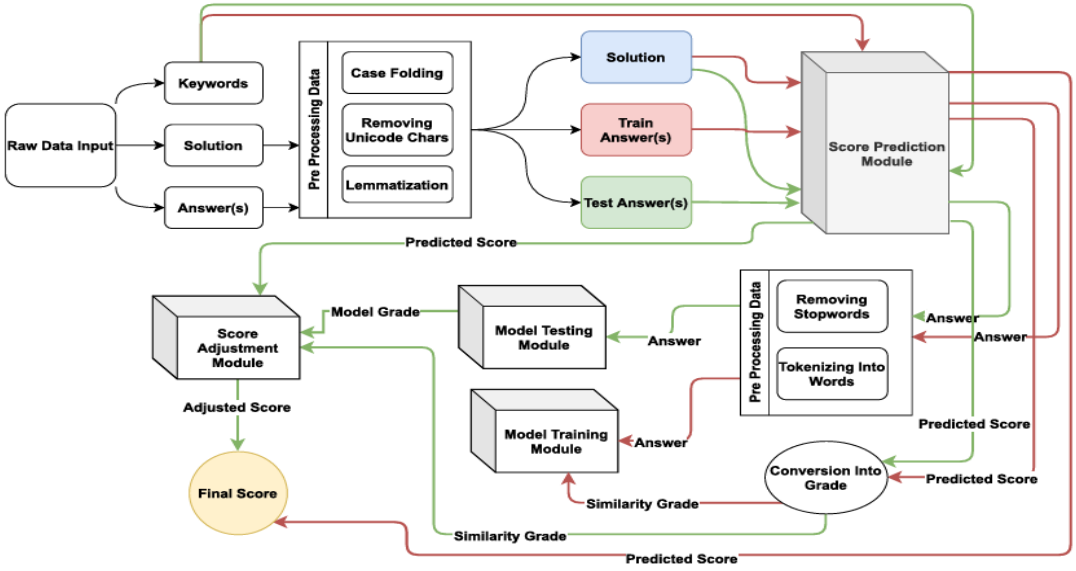

Figure 1.

Process:-

This model consists of Machine learning models trained on the data obtained from the result prediction module. Its working is as follows:

- Input data from Result Prediction Module.
- Preprocess the solution and answer, removing stop words, and use Countvectorizer to represent them in either Bag of Words or TF-IDF form.
- Convert the overall score obtained from Result Predic tion Module into some category. Four categories A, B, C, and D, are used in the paper, representing 1st, 2nd, 3rd, and 4th quarter of a 100. For example, A represents marks from 0 to 25, and B represents 26 to 50.
- The number of categories is kept to a minimum because of the unavailability of the actual dataset. Practically, these categories can be extended to cover smaller score ranges.
- A machine learning model such as Multinomial Naive Bayes, which performs well for multi-class classi ca tion, is chosen.
- The preprocessed answer is used as testing data with the machinelearning modeltopredictits class/category, and that category is checked with the result obtained from Result Prediction Module. This gives us con dence in the predicted result from the model.
- The preprocessed answer is fed into the machine learn ing model along with its label. Moreover, the model is updated according to new data.
- The predicted class is sent to the Final Score Prediction Module along with the solution, answer, and the overall score.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:03/March-2025 Impact Factor- 8.187 www.irjmets.com

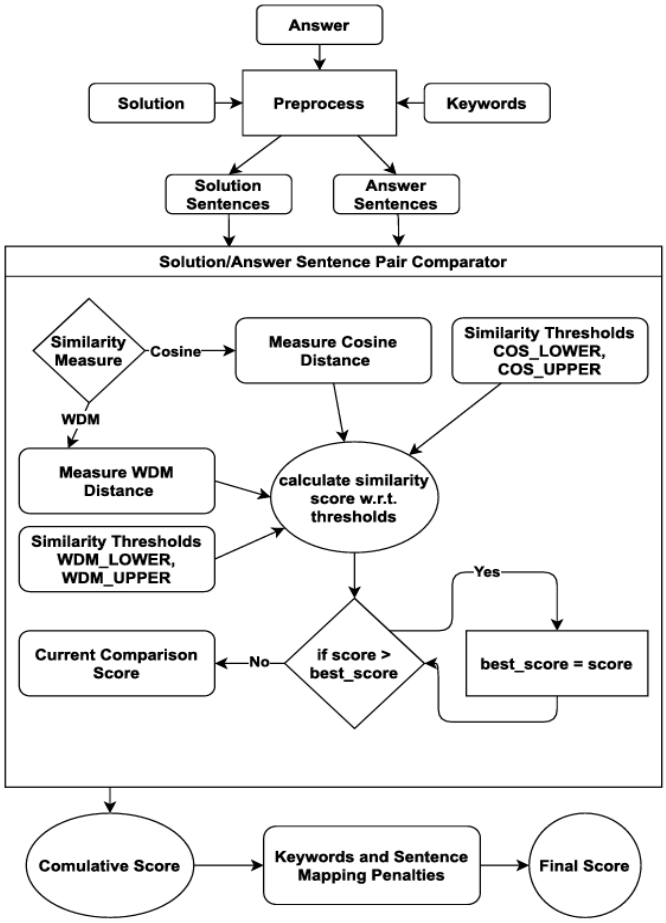

Figure 2. Result Prediction Module

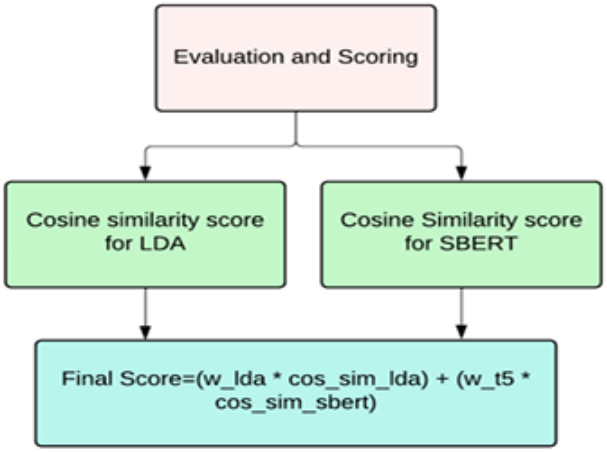

Figure 3. Evaluation and scoring module

IV. CONCLUSION

A novel hybrid model, DAES is proposed in this research for the automatic evaluation of student descriptive answers, representing a significant advancement in educational assessment methodology. By integrating LDA for thematic coverage and T5 for semantic understanding, a versatile and comprehensive approach has been developed which is capable of accurately evaluating student responses across diverse subjects and topics. Through rigorous experimentation and validation, an accuracy of 91%, precision of 91%, recall of 92%, and an F1-score of 91% was achieved, demonstrating the effectiveness and reliability of our model in assessing student performance. The practical implications of our proposed approach have a wide-ranging impact. The capacity to expedite the evaluation process, guarantee consistency and objectivity, and deliver personalized

feedback provides educators with a potent tool for improving teaching and learning results. Furthermore, its scalability and adaptability make it suitable for deployment in a wide range of educational settings, from classrooms to online learning platforms.

While our model has demonstrated strong performance, the importance of ongoing refinement and improvement can be recognized. Future research efforts will focus on expanding the training dataset and exploring additional modalities, which may include consideration of diagrams, mathematical equations, programming code, along with text in answer scripts, to further enhance its accuracy and robustness. A wide range of educational domains and subjects, including but not limited to mathematics, social studies, science, language arts can be implemented for evaluation.

V. REFERENCES

[1] J. Wang and Y. Dong, ‘‘Measurement of text similarity: A survey,’’ Information, vol. 11, no. 9, p. 421, Aug. 2020.

[2] M. Han, X. Zhang, X. Yuan, J. Jiang, W. Yun, and C. Gao, ‘‘A survey on the techniques, applications, and performance of short text semantic similarity,’’ Concurrency Comput., Pract. Exper., vol. 33, no. 5, p. e5971, Mar. 2021.

[3] M. S. M. Patil and M. S. Patil, ‘‘Evaluating student descriptive answers using natural language processing,’’ Int. J. Eng. Res. Technol., vol. 3, no. 3, pp. 1716–1718, 2014.

[4] P. Patil, S. Patil, V. Miniyar, and A. Bandal, ‘‘Subjective answer evaluation using machine learning,’’ Int. J. Pure Appl. Math., vol. 118, no. 24, p. 113, 2018.

[5] J. Muangprathub, S. Kajornkasirat, and A. Wanichsombat, ‘‘Document plagiarism detection using a new concept similarity in formal concept analysis,’’ J. Appl. Math., vol. 2021, pp. 1–10, Mar. 2021.

[6] X.HuandH.Xia, ‘‘Automated assessment system for subjective questions based on LSI,’’ in Proc. 3rd Int. Symp. Intell. Inf. Technol. Secur. Informat., Apr. 2010, pp. 250–254.

[7] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, ‘‘From word embeddings to document distances,’’ in Proc. Int. Conf. Mach. Learn., 2015, pp. 957–966.

[8] C. Xia, T. He, W. Li, Z. Qin, and Z. Zou, ‘‘Similarity analysis of law documents based on word2vec,’ ’inProc. IEEE 19th Int.Conf.Softw.Qual., Rel. Secur. Companion (QRS-C), Jul. 2019, pp. 354–357.

[9] H. Mittal and M. S. Devi, ‘‘Subjective evaluation: A comparison of several statistical techniques,’’ Appl. Artif. Intell., vol. 32, no. 1, pp. 85–95, Jan. 2018.

[10] L. A. Cutrone and M. Chang, ‘‘Automarking: Automatic assessment of open questions,’’ in Proc. 10th IEEE Int. Conf. Adv. Learn. Technol., Sousse, Tunisia, Jul. 2010, pp. 143–147.

[11] G.Srivastava, P. K. R. Maddikunta, and T. R. Gadekallu, ‘‘A two-stage text feature selection algorithm for improving text classification,’’ China Med. Univ., Taiwan, Tech. Rep. 137, 2021.

[12] H. Mangassarian and H. Artail, ‘‘A general framework for subjective information extraction from unstructured English text,’’ Data Knowl. Eng., vol. 62, no. 2, pp. 352–367, Aug. 2007.

[13] B. Oral, E. Emekligil, S. Arslan, and G. Eryiˇ git, ‘‘Information extraction from text intensive and visually rich banking documents,’’ Inf. Process. Manag., vol. 57, no. 6, Nov. 2020, Art. no. 102361.

[14] V. Nandini and P. Uma Maheswari, ‘‘Automatic assessment of descriptive answers in online examination system using semantic relational features,’’ J. Supercomput., vol. 76, no. 6, pp. 4430–4448, Jun. 2020.

[15] H. Tuan Nguyen, C. Tuan Nguyen, H. Oka, T. Ishioka, and M. Nakagawa, ‘‘Handwriting recognition and automatic scoring for descriptive answers in Japanese language tests,’’ 2022, arXiv:2201.03215.

[16] Y. Wu, A. Henriksson, J. Nouri, M. Duneld, and X. Li, ‘‘Beyond benchmarks: Spotting key topical sentences while improving automated essay scoring performance with topic-aware BERT,’’ Electronics, vol. 12, no. 1, p. 150, Dec. 2022.

[17] M. F. Bashir, H. Arshad, A. R. Javed, N. Kryvinska, and S. S. Band, ‘‘Subjective answers evaluation using machine learning and natural language processing,’’ IEEE Access, vol. 9, pp. 158972–158983, 2021.

[18] R. S. Wagh and D. Anand, ‘‘Legal document similarity: A multi criteria decision-making perspective,’’ PeerJ Comput. Sci., vol. 6, p. e262, Mar. 2020.

[19] M. Alian and A. Awajan, ‘‘Factors affecting sentence similarity and paraphrasing identification,’’ Int. J. Speech Technol., vol. 23, no. 4, pp. 851–859, Dec. 2020.

[20] G. Jain and D. K. Lobiyal, ‘‘Conceptual graphs based approach for subjective answers evaluation,’’ Int. J. Conceptual Struct. Smart Appl., vol. 5, no. 2, pp. 1–21, Jul. 2017.

[21] M. Montes-Y-Gómez, A. López-López, and A. Gelbukh, ‘‘Information retrieval with conceptual graph matching,’’ in Proc. Int. Conf. Database Expert Syst. Appl., vol. 1873, Jan. 2000, pp. 312–321.

[22] V. Bahel and A. Thomas, ‘‘Text similarity analysis for evaluation of descriptive answers,’’ 2021, arXiv:2105.02935.

[23] H. Zhang and D. Litman, ‘‘Automated topical component extraction using neural network attention scores from source-based essay scoring,’’ in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, 2020, Art. no. 85698584.

[24] D. M. Blei, A. Y. Ng, and M. I. Jordan, ‘‘Latent Dirichlet allocation,’’ J. Mach. Learn. Res., vol. 3, pp. 993–1022, Jan. 2003.

[25] C. K. Carter and R. Kohn, ‘‘On Gibbs sampling for state space models,’’ Biometrika, vol. 81, no. 3, pp. 541–553, Aug. 1994.

[26] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, ‘‘Exploring the limits of transfer learning with a unified text-to-text transformer,’’ J. Mach. Learn. Res., vol. 21, pp. 1–67, Jan. 2020.

[27] G. Grefenstette, ‘‘Tokenization,’’ in Syntactic Wordclass Tagging. Berlin, Germany: Springer, 1999, pp. 117–133.

[28] K. Sirts and K. Peekman, ‘‘Evaluating sentence segmentation and word Tokenization systems on Estonian web texts,’’ in Proc. 9th Int. Conf. Baltic (HLT), vol. 328, U. Andrius, V. Jurgita, K. Jolantai, and K. Danguole, Eds. Kaunas, Lithuania: IOS Press, Sep. 2020, pp. 174–181.

[29] A. Schofield, M. Magnusson, and D. M. Mimno, ‘‘Pulling out the stops: Rethinking stopword removal for topic models,’’ in Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics (EACL), vol. 2, M. Lapata, P. Blunsom, and A. Koller, Eds. Valencia, Spain: Association for Computational Linguistics, 2017, pp. 432–436.

[30] M. Çagatayli and E. Çelebi, ‘‘The effect of stemming and stop-word removal on automatic text classification in Turkish language,’’ in Proc. 22nd Int. Conf. Neural Inf. Process. (ICONIP) (Lecture Notes Computer Science), vol. 9489, S. Arik, T. Huang,W.K. Lai, and Q.Liu,Eds. Istanbul, Turkey: Springer, 2015, pp. 168–176