Automating Assessment with NLP-Powered Answer Checker

Akshada Tari[1]

[1]*Student, Department of Information Technology and Engineering, Goa College of Engineering, Farmagudi, Goa, India*

---***---

Abstract - *In the ever-evolving landscape of education, the demand for efficient and accurate assessment tools has grown exponentially. This paper explores the integration of Natural Language Processing (NLP) techniques to develop an Automatic Answer Checker (AAC) with a focus on enhancing the grading process. NLP, a subfield of artificial intelligence, offers a powerful set of tools for understanding and analyzing human language, making it an ideal candidate for automating the assessment of the responses. The proposed AAC system employs advanced NLP ML algorithms to evaluate and grade answers submitted by students. The system aims to streamline the grading process. As education transitions into a more technologically-driven era, the integration of innovative tools like the AAC system holds the potential to enhance the educational experience, ensuring fair and timely evaluation while promoting a deeper understanding of the subject matter.*

Key Words: **Natural Language Processing, Cosine Similarity, Sentence Transformer, Automatic Answer Checker**

1.INTRODUCTION

As the landscape of education evolves, the adoption of online tests and examinations has gained widespread popularity, aiming to alleviate the burdens associated with traditional examination evaluation processes. While online assessments typically focus on objective or multiple-choice questions, the evaluation of subjective-based questions and answers has proven challenging due to the complexities and inefficiencies inherent in the grading process.

Recognizing this limitation, there is a growing demand for innovative solutions that address the assessment of the responses in online exams. In response to this need, our focus turns to the development of an Automatic Answer Checker (AAC) model, designed to evaluate the answers and provide grading. This model incorporates advanced Natural Language Processing (NLP) techniques to comprehend, analyse, and assign weightage to the responses, bridging the gap between traditional human grading and the efficiency of online evaluation.

This paper explores the potential of the AAC model to revolutionize the assessment paradigm, offering a solution that ensures the accuracy and reliability of grading for the questions in the modern era of education. The subsequent overview delves deeper into the key components and functionalities of this innovative approach, shedding light on how NLP contributes to the efficiency and effectiveness of the grading process.

The rest of the paper is organized as follows: The method proposed which includes sentence transformer, All-mpnet-base-v2, Cosine Similarity is given in Section 2. The Literature survey is covered in Section 3. Conclusions are discussed in Section 4.

2. Related Work

[1]. Jagadamba G and Chaya Shree G proposed Artificial Intelligence-based answer verifier to do the job of examiner/evaluator. Artificial Intelligence-based Answer Verifier calculates the score of the student by combining various parameters such as keywords, proper grammar. The value of keywords ranges from 1 to 6 where 1 is for Excellent and 6 is for Very Poor. The values of "grammar" attribute ranges from 0 which is for Improper and 1 which is for Proper. The system is more efficient for answers that are point to point. Provides overall accuracy of 80%. The module is designed and tested for the 'Cosine Similarity' algorithm. Cosine Similarity is used to measure the similarity between two non-zero vectors which are the inner product space. The measure is the cosine of the angle between the two vectors i.e., 0° is 1, and less than 1 for any angle in the interval $(0, \pi)$ radians

[2]. Shreya Singh , Prof. Uday Rote , Omkar Manchekar ,Prof. Sheetal Jagtap ,Ambar Patwardhan, Dr. Hariram Chavan proposed the concept of Artificial Intelligence, OCR, and NLP to solve the problem. The answer sheets of the student is compared to the model answer sheet by the evaluator and will then generate the final score based on multiple parameters(sentence splitting, Jaccard similarity, grammar checking and sentence similarity). For the implementation of the system: cosine similarity and Jaccard similarity was used. The major setback of cosine similarity is it takes into consideration even the repetition of the same words. The measure of cosine similarity is higher primarily due to considering the repetitive similar words multiple times. This can generate a greater similarity level completely based on the number of times the word is repeated. Hence, Jaccard similarity is the better measure of similarity for the system.

[3].Potsangbam Sushila Devi, Sunita Sarkar, Takhellambam Sonamani Singh, Laimayum Dayal Sharma, Chongtham Pankaj and Khoirom Rajib Singh proposed a system designed to evaluate and check identical and semantic related answers

with the target answer by providing the similarity percentage. The evaluation of subjective answers are carried out using Bidirectional Encoder Representation Transformers (BERT) model to understand related answers with the target answers. BERT has the capability to understand language and next prediction sentence by its pre-trained model. The BERT model is used to generate the vectors for each word in a sentence and pooling method is applied to reduce the spatial matrix into vector for each sentence. The cosine method is used to compare the target answers and related answers. The result is evaluated to be considered for subjective answers correction approach.

[4]. Rishabh Kothari, Burhanuddin Rangwala, Kush Patel proposed the study that employs Machine Learning (ML) and Natural Language Processing (NLP) to automate grading. By comparing student answers with ideal ones provided by exam authorities, a similarity score is generated and mapped to grades. The study offers a Django-based web application for online exams, providing near real-time grading. Administrators have the flexibility to adjust grades as needed, recognizing that no ML model is perfect.

[5]. Ragasudha, M.Saravanan proposed a system for secure automatic question paper generation and subjective answer evaluation. Admins create a question database categorized by Bloom's taxonomy, and with a click, questions are generated. Cryptography ensures secure delivery of papers. Answer evaluation is automated using keyword matching algorithms. This system reduces human error and saves time in both question generation and answer evaluation.

[6]. Ashutosh Shinde, C. Nishit Nirbhavane, Sharda Mahajan, Vikas Katkar, Supriya Chaudhary proposed an ideal solution would involve a software application with a comprehensive database of questions, corresponding answers, and assigned marks. This application is capable of verifying user responses by comparing them with the provided answers. By employing such a system, evaluators can significantly reduce their workload and increase efficiency.

3. Approaches to Measure Answer Similarity in NLP

In the realm of Natural Language Processing (NLP), there are several methods used to check answer similarity, often drawing inspiration from techniques in information retrieval, machine learning, and linguistics.

Some of the prominent approaches are: cosine similarity, Word Embeddings, BERT-Based Methods, Semantic Text Similarity Models, Graph-Based Methods, Edit Distance.

These methods provide a broad spectrum of approaches to measuring answer similarity in NLP, each with its advantages and suitable applications. Depending on the nature of the data and the specific requirements of the task, one or a combination of these methods may be more appropriate.

3.1 Cosine Similarity

Cosine similarity measures the cosine of the angle between two vectors in a multi-dimensional space. In NLP, each answer can be represented as a vector, typically using techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings. Cosine similarity is widely used because it's efficient, computationally inexpensive, and insensitive to the vector's magnitude.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

3.2 Word Embeddings

Word embeddings represent words as dense, continuous vectors in a high-dimensional space, where the distance and direction between vectors encode semantic information. These embeddings are typically generated using unsupervised learning techniques like Word2Vec, GloVe, or fastText. By averaging or summing the embeddings of words in each answer, you can compare their similarity using cosine similarity or other distance metrics.

Word embeddings themselves are generated through training algorithms like Word2Vec or GloVe, and there's no specific formula for computing similarity, although cosine similarity is often used.

3.3 BERT-Based Methods

BERT (Bidirectional Encoder Representations from Transformers) and its variants are powerful pre-trained models for various NLP tasks. Fine-tuning BERT for answer similarity tasks or extracting embeddings from BERT outputs can effectively capture semantic similarity. BERT is trained on large corpora with tasks like masked language modeling and next sentence prediction, enabling it to capture rich contextual information. Fine-tuning BERT involves training it on a specific downstream task (e.g., answer similarity) with appropriate loss functions.

3.4 Semantic Text Similarity Models

These models are specifically trained to measure the similarity between two text inputs by capturing fine-grained semantic information. Models like Siamese-CNN (Convolutional Neural Networks), Siamese-LSTM (Long Short-Term Memory), or transformer-based architectures (e.g., Siamese-BERT) are commonly used. These models learn to generate embeddings for text inputs in such a way that similar inputs are mapped closer together in the embedding space. They involve training neural network architectures to learn embeddings that capture semantic similarity.

3.5 Graph-Based Methods

Graph-based methods represent text as a graph where nodes are words or phrases, and edges represent relationships (e.g., syntactic or semantic). Similarity is computed based on graph properties like shortest path distance, graph kernel, or node embeddings. These methods are particularly useful when capturing structural relationships between words or phrases.

Graph-based methods utilize algorithms for graph comparison, such as graph edit distance or graph similarity measures, which involve complex graph analysis but don't have single formulas.

3.6 Edit Distance

Edit distance measures the minimum number of operations (e.g., insertions, deletions, substitutions) required to transform one answer into another. It's simplistic but can be effective for measuring similarity in certain contexts. Variants like Levenshtein distance, Damerau-Levenshtein distance, or Jaccard similarity are commonly used.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

The formula varies depending on the specific variant of edit distance being used, such as the dynamic programming-based algorithm for Levenshtein distance.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

4. Proposed Method

4.1 Sentence Transformer

The Sentence Transformer library provides an efficient way to obtain contextualized embeddings for sentences using pre-trained transformer models. These embeddings capture rich semantic information, allowing for accurate comparison and analysis of sentence meanings.

It simplifies the process of working with transformer models, making it accessible to users without extensive knowledge of deep learning techniques. Additionally, it offers flexibility in choosing from a variety of pre-trained models to suit different applications and requirements.

Sentence Transformer is suitable for tasks such as similarity comparison, semantic search, paraphrase detection, and more, making it an excellent choice for automatic answer checking.

4.2 all-mpnet-base-v2

The "all-mpnet-base-v2" model is a specific pre-trained transformer model provided by the Sentence Transformer library. It is based on the MPNet architecture, which is known for its effectiveness in capturing contextual information and generating high-quality embeddings.

This model is expected to provide robust contextual embeddings for sentences, enabling accurate assessment of the semantic similarity between reference and student answers. Its usage within the Sentence Transformer framework ensures compatibility and ease of integration.

MPNet-based models have demonstrated strong performance in various NLP tasks, making "all-mpnet-base-v2" a reliable choice for generating sentence embeddings.

4.3 Cosine Similarity

Cosine similarity is a metric commonly used to measure the cosine of the angle between two vectors. In the context of sentence embeddings, it assesses the similarity in direction between the vectors, indicating how similar the two sentences are in meaning.

Cosine similarity provides a straightforward and intuitive way to compare the similarity of sentence embeddings. A cosine similarity score of 1 indicates identical vectors (perfect similarity), while a score of 0 signifies orthogonal vectors (no similarity).

The cosine similarity score can be easily interpreted, making it suitable for providing feedback to users on the similarity between reference and student answers. It also allows for thresholding to determine whether the answers are sufficiently similar or not.

5. Proposed Model

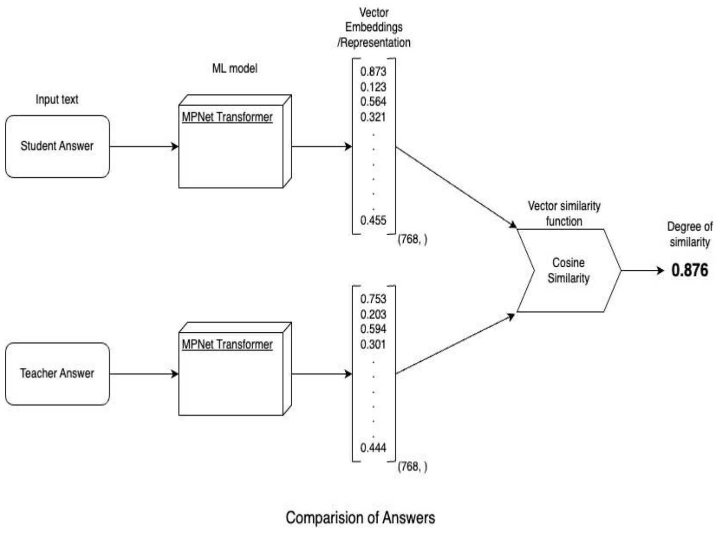

Fig -1: Comparison of Answers

To compare the answers, a MPNet based model can be used to convert the student answer and teachers answer into vector embeddings of size 768 each and further processed using cosine similarity to find out degree of similarity.

6. CONCLUSIONS

In this paper, we explored methods to check the correctness of the students answer and teachers answer using various NLP techniques. Additionally, the paper also proposes an approach using sentence transformer.

The paper results its potential to streamline grading processes, relieving educators of manual efforts. Overall, the paper showcases the viability of NLP-driven automated grading systems, opening avenues for continued innovation in educational technology.

REFERENCES

[1] Jagadamba G and Chaya Shree G " Online Subjective answer verifying system Using Artificial Intelligence." Proc of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC),IEEE 2020

[2] Shreya Singh , Prof. Uday Rote , Omkar Manchekar ,Prof. Sheetal Jagtap ,Ambar Patwardhan,Dr. Hariram Chavan,"Tool for Evaluating Subjective Answers using AI (TESA)" ,©2021 IEEE

[3] Potsangbam Sushila Devi, Sunita Sarkar, Takhellambam Sonamani Singh, Laimayum Dayal Sharma, Chongtham Pankaj and Khoirom Rajib Singh-"An Approach to Evaluating Subjective Answers using BERT model," 2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT) | 978-1-6654-9781-7/22/$31.00 ©2022 IEEE

[4] Rishabh Kothari ,Burhanuddin Rangwala ,Kush Patel "Automatic Subjective Answer Grading Software using Machine Learning" 7th International Conference on Trends in Electronics and Informatics (ICOEI) ©2023 IEEE 2023 | DOI:10.1109/ICOEI56765.2023.1012578

[5] Ragasudha, M.Saravanan "Secure Automatic Question Paper Generation with the Subjective Answer Evaluation System" International Conference on Smart Technologies and Systems for Next Generation Computing(ICSTSN) | ©2022 IEEE|DOI10.1109/ICSTSN53084.2022.9761323

[6] Ashutosh Shinde, C. Nishit Nirbhavane, Sharda Mahajan, Vikas Katkar, Supriya Chaudhary, "AI Answer Verifier," Int. Journal of Engineering Research in Computer Science and Engineering (IJERCSE), Volume 5, Issue 3, March 2018.

[7] P. Pisat, D. Modi, S. Rewagad, G. Sawant and D. Chaturvedi, "Question paper generator and answer verifier," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017, pp. 1074-1077, doi: 10.1109/ICECDS.2017.8389603.

[8] S. W. Yeh, L. K. Wang, and H. W. Cheng,"Automated short answer grading using machine learning techniques", IEEE International Conference on Information Reuse and Integration (IRI), 2017

[9] D. Kuznetsov, Y. S. Kolesova, and E. V. Kolesova, "Automatic Evaluation of Students' Answers Based on Word Embeddings" IEEE International Conference on Advanced Learning Technologies (ICALT), 2018

[10] S. Yuan, W. Zhu, and D. Luo, "Automated Short Answer Grading Using Sentence Similarity based on Pretrained Word Embeddings", IEEE International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), 2019

[11] T. T. Nguyen, H. L. Nguyen, and Q. V. Nguyen, "Automated Short-Answer Grading with BERT"IEEE Access, 2020

[12] S. Kumari and A. L. Sharma,"Automatic Short Answer Grading using Deep Learning Techniques" IEEE International Conference on Computing, Communication, and Automation (ICCCA), 2021

[13] S. M. Rahaman, R. T. P. Sooriamurthi, and C. E. Hughes, "Automated grading of short-answer questions using combination of feature-based techniques and neural networks", IEEE Transactions on Learning Technologies, 2018

[14] S. B. Bhagwat and S. S. Mantha, "Automated Assessment of Students' Answers to Open-ended Questions", IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2019

[15] P. A. Deokar and V. M. Thakare, "Automated Short Answer Grading using Machine Learning Techniques: A Survey", IEEE International Conference on Computer and Communication Technology (ICCCT), 2020

[16] T. C. Loo, M. H. Ng, and S. K. Adhikari, "Automated Answer Grading for Short Text Answers Using LSTM and CNN" IEEE Access, 2022