Information Retrieval Mid-Project Review

Project Title and Details of the Group:

"Automatic Answer Evaluation System"

Project group No. - 25

Team members: -

- Ankit Kumar ankit18218@iiitd.ac.in
- Prashant prashant18360@iiitd.ac.in
- Udbhav Gupta udbhav17319@iiitd.ac.in
- Sudhir Kumar sudhir18268@iiitd.ac.in
- Ram Kumar ram18080@iiitd.ac.in

Motivation and precise problem statement:

Traditional answer evaluation, i.e. manual checking of answers, takes a lot of time and energy. However, objective type questions can be evaluated using computers very efficiently. Still, when it comes to theoretical evaluation of answers, there does not exist a platform for checking the answers very precisely and efficiently. There is always a need for a checker to check the answers manually, which takes a lot of effort and time, and sometimes he/she has to provide the feedback as well, which would again take a lot of effort and analysis.

Hence, we propose a system that evaluates the assessment copies automatically according to the rubric set by the checker for every type of question (subjective and objective) and multiple types of documents like pdf, scanned pdf. Our system would also give feedback over every question in a sheet generated for each student.

How is our proposed system better?

- Checking copies by teachers could lead to human error, but our system shows uniformity over all copies, which reduces human errors.
- Our system could reduce lots of human time, power and money, which could be utilized in a better place.
- Our system would also accept the handwritten scanned type file, which students could submit.
- Our system provides feedback over every solution to the student in a .csv file and also collective feedback of the whole class for each question in the form of a pie chart and histogram to the checker.
- Our system is also supposed to provide the partial marking or binary marking option, which could be opted by the checker.

Literature review:

Answer Evaluation Using Machine Learning:

We have to scan the answer to a question; then, the system will automatically generate the keyword using the OCR technique. Based on keywords written in the solution and the keywords in the dataset, the application provides marks.[1]

Limitation-

This model can not evaluate the handwritten text from the answers. It can only check the printed text from the image of answers. It does not include a feedback feature, as well.

ONLINE SUBJECTIVE ANSWER CHECKER:

The answer selected by the student (String1) is compared with the answer stored in the database(String2) provided by the checker. If String1= String2 then score=+1 or else score=0. After processing the results of the test into the database, the results are displayed to the student.[2]

Limitation-

This system is not efficient, but it is less time consuming and still lacks the feature of extracting the handwritten text from the image, i.e. this model checks only digital or printed text.

It does not include a feedback feature, as well.

NATURAL LANGUAGE PROCESSING AND ARTIFICIAL NEURAL NETWORKS--

Once the student has submitted the answer, this system will automatically calculate the result using two NLP algorithms (Natural Language Processing) and ANN (Artificial Neural Network). Here this model uses the Artificial Neural Networks algorithm for the standard answer comparison and evaluates the same answer using the Natural language processing [NLP] algorithm for grammar mistakes and stores the marks in the student database. Some fundamental linguistic analysis is performed in a natural language parser and is used to perform POS tagging of the student's answer text. After linguistic analysis, the student's answer text is processed by the artificial neural networks algorithm; it will compare the student's answer text with the checker's answer and with keywords. Each process's result is calculated using a "marks calculator" to compute the total marks obtained by the student for his/her answer and finally compares both marks and provides a final result.[3]

Limitation-

This model is still in the development phase for recognition of handwritten text using deep learning methods, i.e. it is still not implemented.

Automated Answering for Subjective Examination [4]

This paper is focusing on the inference process, which required for developing such type of

systems. This system can assess word and one sentence based answers with more than 80% of efficiency. While answering single sentence answers, paraphrasing is considered for assessing the variations occurring due to vocabulary use.[4]

Limitation: Although it was a good platform, it does not provide feedback on each answer and does not work over handwritten text.

AI Answer Verifier [5]

This is an automatic answer checker application that checks and marks written answers similar to a human being. This software application is built to check subjective answers in an online examination and allocate marks to the user after verifying the response. The system requires you to store the original solution for the system. This facility is provided to the admin. The admin may insert questions and respective subjective answers in the system. These answers are stored as notepad files. When a user takes the test, he is provided with questions and areas to type his answers. Once the user enters his/her answers, the system then compares this answer to

The original answer is written in the database and allocates marks accordingly. Both the answers need not be exactly the same word to word. The system consists of inbuilt artificial intelligence sensors that verify answers and allocate marks accordingly as good as a human being.[5]

Limitation:

It was friendly for the instructor, but if an answer sheet has many types of questions and if some have partial marking and some have binary marking, then this model won't work perfectly.

Supervised Word Mover's Distance [6]

The WMD elevates high-quality word embeddings to a document metric by formulating the distance between two documents as an optimal transport problem between the embedded words. However, the document distances are entirely unsupervised and lack a mechanism to incorporate supervision when available. To overcome this, an efficient algorithm that is supervised has been proposed in this paper called Supervised-WMD (S-WMD) metric.

The supervised training minimizes the stochastic leave-one-out nearest neighbour classification error on a per-document level by updating an affine transformation of the underlying word embedding space and a word-importance weight vector. This algorithm provides an arbitrarily close approximation that results in a practical and efficient update rule.

From Word Embeddings To Document Distances [7]

It is based on the Word Mover's Distance (WMD) algorithm. The WMD distance measures the dissimilarity between two text documents as the minimum amount of distance that the embedded words of one document need to "travel" to reach the embedded words of another document. The WMD metric leads to unprecedented low k-nearest neighbour document classification error rates.

We will first make each dataset and a set of classic and state-of-the-art document representations and distances. Now we will try to compare the nearest neighbour performance of WMD, and then we will apply the best competing methods on these datasets. Finally, we examine how the fast lower bound distances can speed up nearest neighbour computation by prefetching and pruning neighbours.

Proposed method:

We have divided the whole project into the following five subtasks, which are mentioned below.

1. User Interface:

Here We will make the visual interface of the entire system look representable and usable.

- Upload rubric,
- Set number of questions checker wants to evaluate,
- Uploading the directory which contains a database of student responses, here the domain of files are handwritten scanned file or typed text simple files (.docx, .text)
- After completion of the evaluation, a pop-up option will be available for downloading the .csv file, which contains all the marks and feedback of each student.

2. Text Retrieval:

- In this section, we will get the rubric and the database of the students by applying different types of indexing and searching methods to make the system work fast.
- Here extraction from rubric is that answer of a particular question, type of marking, their marks set by the checker. Partial marking will be fixed by the checker in the rubric(checker needs to write the marks of the particular answer), for example, if it is a five marks answer, for partial marking [(2 + 2 + 1) marks] and for binary marking [(5) marks]. This part will extract this information from a particular answer.
- And then, it will retrieve the particular answer efficiently that is required for the evaluation process.

3. Reading of image or Scanned files.

- Here we have an option to run our algorithm based on the type of file. If we get the scanned image, we will try to fetch the information using architecture for handwritten text recognition, which is supposed to be more reliable and fast as compared to OCR.

4. Evaluation of Answer

- Whether the question is Subjective or Objective, we will run the different algorithms.
- For the Objective answer, we will use the exact matching algorithm.
- For a Subjective answer, we will use the concepts of word embedding. In this method, we will make the corpus from the rubric answer and use the Word2vec technique in which each word has n-dimension (like 50 or 100 dimensions), and then calculate the similarity distance (which is the word mover's distance) between rubric corpus vector and each student answer.
- Then we will produce the .csv file for the output and finally will get the evaluated database.

5. Feedback of Answer

- Here we will provide feedback for each answer of the student so that they can understand the mistake, and it will also make it easy for the student as well as the professor to understand their weak area.

- We will set some attributes like(answer length relative to rubric answer length, specific keywords matched or not, grammatical errors, marks of an answer) then calculate the attribute score out of 10. We also set some 15-20 generalized feedback templates in the preprocess work. After calculating the attribute score of each answer, then our system will retrieve the most relevant feedback from the template according to the attribute score.
- These student's individual feedback will be stored in a .csv output file.
- We will provide overall or collective feedback from all student's answer feedback, which will help the professor know how many students made a similar kind of mistake.

Reference:

[1] S. Bharadia, P. Sinha, and A. Kaul, *Answer Evaluation Using Machine Learning*. 2018.
[2] M. Mathew, A. Chavan, S. Baikar, ONLINE SUBJECTIVE ANSWER CHECKER, *International Journal of Scientific & Engineering Research, Volume 8, Issue 2, February-2017, ISSN 2229-5518*
[3] S. M. Patil and S. Patil, “Evaluating Student Descriptive Answers Using Natural Language Processing,” *Int. J. Eng. Res.*, vol. 3, no. 3, p. 3, 2014.
[4] Asmita Dhokrat, Gite Hanumant, C. Namrata Mahender “Automated Answering for Subjective Examination” in International Journal of Computer Applications · October 2012.

[5] Sakshi Berad, Pratiksha Jaybhaye, Sakshi Jawale “AI Answer Verifier”.e-ISSN: 2395-0056 Volume: 06 Issue: 01 | Jan 2019 p-ISSN: 2395-0072

[6] Gao Huang, Chuan Guo, Matt J. Kusner, Yu Sun, Kilian Q. Weinberger, Fei Sha, Supervised Word Mover's Distance 2016.

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

[7] M. Kusner, Y. Sun, N. Kolkin and K. Weinberger, "From Word Embeddings To Document Distances", PMLR, 2021. [Online]. Available: http://proceedings.mlr.press/v37/kusnerb15.html. [Accessed: 22- Mar- 2021]