RESEARCH ARTICLE OPEN ACCESS

Subjective Answer Evaluation system

Alok Kumar [1], Aditi Kharadi [2], Deepika Singh [3], Mala Kumari [4]

Department of Computer Science and Engineering, CSJM University - Kanpur

ABSTRACT

This Automated text scoring (ATS) or subjective-answer evaluation is one of the big hurdles in the technical advancement of academics. Reading each answer meticulously and scoring them impartially becomes a monotonous task for many in the teaching profession, especially if the answers are long. Another big challenge is comprehending the student's handwriting. Marking criteria may also vary largely with domain, for instance, credit is given to usage of correct grammar in some cases, while other domains may require certain keywords to be present in student's answers. In this paper, we have tried to approach this problem with three perspectives- two standard linguistic approaches and a deep learning approach. The first approach employs the presence of certain keywords as a marking criteria and also includes a handwriting recognizer that can extract text from scanned images of the handwritten answers. The second approach uses similarity between an understudy and a benchmark answer. This paper also proposes the use of a sequential model, which is trained on the Automated Student Assessment Prize - Automated Essay Scoring (ASAP-AES) dataset for evaluating long answers.

Keywords :- Natural Language Processing (NLP), Cosine Similarity, Jaccard Similarity, Synonym Similarity, Bigram Similarity, Sequential Model, LSTM, Root Mean Squared Error (RMSE)

I. INTRODUCTION

This document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

The history of automated answer evaluation is quite long. Currently, the objective-answer evaluation systems are abundant, but it is not the same for subjective-answer evaluation. Manual answer evaluation is a very time consuming job. Not only this but it also requires a lot of manpower. Because of the obvious human error, it can sometimes be partial to few students, which is not preferred. So our system will evaluate answers using three different approaches. The motivation behind using three different approaches is to get best results in every possible domain. As all are aware that different domains require different bases for the evaluation process. For an answer written on some event of history, it is required for it to have some necessary keywords like date, place or name which is not the case with essays or other domains where main focus is on the absolute meaning.

Answer evaluation or ATS is the task of scoring a text using a set of statistical and NLP measures or neural networks. Some domains may prefer the quality of answer to be the scoring criteria which highly depends on the stop words(or keywords) present in the answers. The first approach used by us revolves around the keywords present in the answer and the keywords expected to be present in the answer. It counts the matched keywords and uses it along with the length of the answer to generate the final score. The second approach simply measures the similarity score of the understudy answer when compared with the model answer. These similarity measures are cosine similarity, jaccard similarity, synonyms similarity and bigram similarity. These scores are combined together to get the final score for the answer. The third approach is mainly used to evaluate long answers using a sequence-to-vector model with stacked layers trained on ASAP-AES[22] dataset. We have elaborated the methodologies of all three approaches in section 3. The system evaluation has been done in section 4. The final results and conclusion-future work are elaborated in sections 5 and 6 respectively.

II. RELATED WORKS

This document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

Many researchers have proposed influential and novel approaches for the task of ATS. One of the earliest essay scoring systems was proposed in Project Essay Grade[1], which used linear regression over the vector representations of the answer/text for scoring. Patil et al.[2] suggest usage of a pure linguistic approach for scoring subjective answers in text format after extracting them from scanned images of the handwritten answers. Many researchers have looked at ATS as a supervised text classification task(Rudner et al.[4], Sakaguchi et al.[5]) Landauer et al.[3], for example, proposed

usage of latent semantic analysis and similarity measures for scoring a text. Even though standard linguistic approaches have given substantial performance over text scoring, they still need to get free of domain specificity.

Impedovo et al.[6] proposes the use of optical character recognizer(OCR) for handwriting Extraction. And also discusses the problems in recognizing handwritten and printed characters.An advanced version of OCR model is proposed by Vanni et al.[9]. It uses an artificial neural network at the backend to get high accuracy. Nagy et al. [7] also discusses the OCR model and its strengths and weaknesses. Pradeep et al.[8] uses a new approach called diagonal based feature extraction by training a neural network using many handwritten alphabets. Shi el al.[10] discusses the scene text recognition and checks the performance on both lexicon-free and lexicon- based recognition. The study of Oganian et al.[15] gives insight on the bilingual word recognition system.

The study of Lahitani et al. [17] has been a source of motivation for our second approach. They use similarity scores for the answer and generate the final score. Bluche et al. [18] used LSTM-RNN to predict open bigram and performed experiments on public databases, Rimes and IAM. Nau et al. [19] have combined latent semantic analysis and linear regression to predict the score of answers.

With advances in deep learning, such systems have surpassed the past benchmarks in terms of performance. A paper by Tai et al.[11] proposes usage of Tree-LSTM instead for general LSTMs (linear chain structure), for semantic representation of text. The basic thought is that any natural language combines its words with phrases. This custom LSTM structure outperforms in the task of predicting semantic relatedness of two sentences. Another paper by Tang et al.[12] proposed learning the sentiment-specific semantic representation for the analysis of the entire document. Alikaniotis et al.[13] proposed usage of LSTMs to represent semantics of the understudy answer and word representation model that learns the impact of a word on the text's score. Laxmi et al.[14] propose usage of ANN for comparing the understudy answer with a benchmark answer and keyword list. The same answer is also evaluated using a NLP system for deducing marks over grammatical/spelling mistakes. The scores predicted by the two systems are compared and a final score is calculated.

The Hewlett Foundation sponsored a contest on Kaggle in 2012 named "Automated Student Assessment Prize"(ASAP). They released a dataset, which has been used by many text scoring systems(Shermis, [16]).

III. METHODOLOGY

Proposed system involves 3 approaches viz. keyword matching, similarity measures and sequential model, each of which are elaborated in sections 3.1, 3.2 and 3.3.

3.1 Answer Evaluation using the keyword matching technique.

For domains that include many pre-specified words and figures, scoring must be based on the presence of that information in the answer. This is where the keyword matching criteria comes into picture. The flow process of this approach is shown in figure 1.

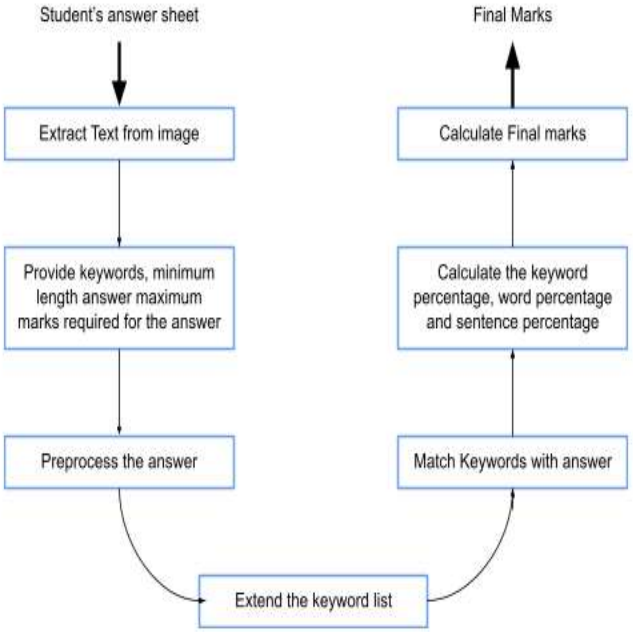

Figure 1: An overview of answer evaluation using keyword matching technique

A. Providing Inputs

Before starting with the evaluation process, we need to provide our system with some inputs to get accurate and fair marking for the students' answers. These inputs are:

a) Students Answer: The most obvious input here is the students answer which is to be evaluated based on some predefined rules. This can be in the form of a scanned image or a text, as suitable.

b) Keywords List: This list consists of the keywords that are expected to be present in the student answer. These keywords are related to the domain of the question of which answers are being evaluated.

c) Maximum Marks: This is the maximum marks out of which the student answer is to be marked.

d) Minimum Length: This is the minimum length of the answer above which the answer will be considered a good answer, and no marks will be deducted if a student's answer exceeds this limit. This number depends on the maximum marks of the answer.

e) Maximum Matching Keywords: Expecting the student to mention all the keywords from the provided keywords list in his/her answer is not practical, so we provide a

minimum number of keywords which is expected to match with the list for the answer to score good marks.

B. Text Extraction using Google vision API.

The student answer is provided to the system in the form of an image, which is the scanned image of the student's answer sheet. This answer extracting function increases the use of the answer evaluation system in real life. In this step, input image is provided to the system. Then the text written on the answer sheet is extracted using vision API[20]. Sample input is shown below in figure 2.

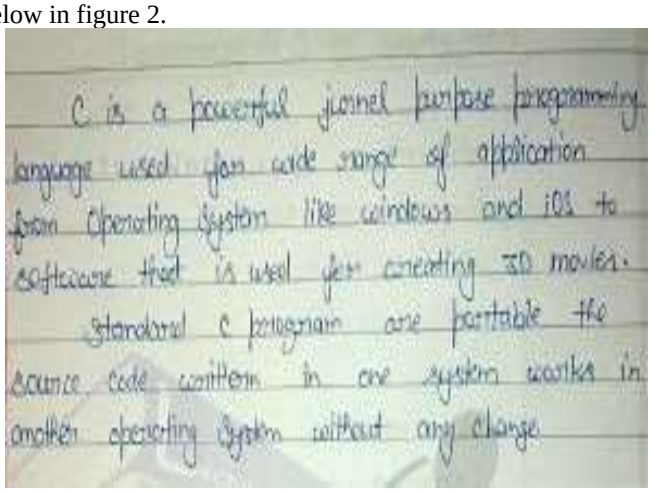

Figure 2: Input to Handwriting Extractor

Text extracted from input is as-

C is a powerful general purpose programming language used for a wide range of applications from operating systems like windows and ios software that is used for creating 3D movies. Standard C programs are portable the source code written in one system works in another operating system without any change.

C. Preprocessing

The input answer is word tokenized and converted into a list of words. The number of sentences and the number of words present in the answer plays an important role in marking the question based on a given maximum mark. So, we find these parameters, which will be used in calculating the final score of the answer. The last step of pre-processing is to convert the words into lowercase and eliminate special characters, punctuation marks and stop words from the obtained list.

D. Extending the Keywords List

The words present in the keyword-list provided by the user may have some related form, which is not mentioned in the list. So the list is further extended to get a dense collection of words for fair evaluation of the answers.

Let a keyword present in the list be "execute", then the words like ['execution', 'executes', 'executing', 'executable'] should be added to the list for fair marking. The list is extended using a lexical database 'Wordnet'.

Each word present in the list is matched in the 'Wordnet' database and if it is present in the database, all the words related to that word are also added to the list.

E. Keyword matching

After extending the keywords list, all the words in the preprocessed student's answer are matched against the extended keyword list. The numbers of matching keywords are counted for further calculations.

F. Calculating percentages

The required percentages are keyword percentage, word percentage, and sentence percentage. The weight of these parameters are different, which is calculated by manually evaluating answers and checking the role of these parameters in the final score. The weights are derived from the combined observation of Kapoor et al.[28], Mahmud et al.[29], and Bharadia et al.[30]. The weights shown in equations 1,2,3.

$$\text{keywords}_{\text{percentage}} = 0.65 \times \frac{\text{keywords\_matched}}{\text{expected\_keywords}} \quad \ldots. (1)$$

$$\text{word}_{\text{percentage}} = 0.25 \times \frac{\text{no\_of\_words}}{\text{expected\_no\_of\_words}} \quad \ldots. (2)$$

$$\textit{sentence}_{\textit{percentage}} = 0.10 \times \frac{\text{no\_of\_sentences}}{\text{expected\_no\_of\_sentences}} \quad \ldots..(3)$$

G. Final Marks Calculation

Final marks are calculated by simply combining the above defined percentages, using the formula stated in equation 4.

$$\text{total}_{\text{marks}} = \text{maximum}_{\text{marks}} \times \left(\text{keywords}_{\text{percentage}} + \text{word}_{\text{percentage}} \text{ and } \text{sentence}_{\text{percentage}} \right) \quad \ldots (4)$$

3.2 Answer Evaluation using similarity measures

Evaluators in general keep a model answer by their side to rate a student's answer based on that benchmark answer. A similar approach has been followed in this perspective of answer evaluation. A benchmark answer and an understudy answer are feeded to the system, which calculates the similarities between the two in various domains. The flow of the process is shown in figure 3.

A. Text Preprocessing:

The under study answer and the benchmark answer, both are preprocessed before checking similarity between them. The preprocessing here is similar to the preprocessing defined in the first approach. The answer is word tokenized and all the punctuation marks and stopwords are removed. Since we do not have a keyword list to have all the forms of the words, here word lemmatization is performed.

Word lemmatization means converting the word into its base form. For example, words executing, executed, executes, etc. are converted to its base form: execute.The last part in preprocessing is the removal of duplicates to get two clean lists of words for further processing.

B. Measure Similarities:

Now we have two preprocessed lists of words, one obtained from an understudy answer and another one from the benchmark answer.

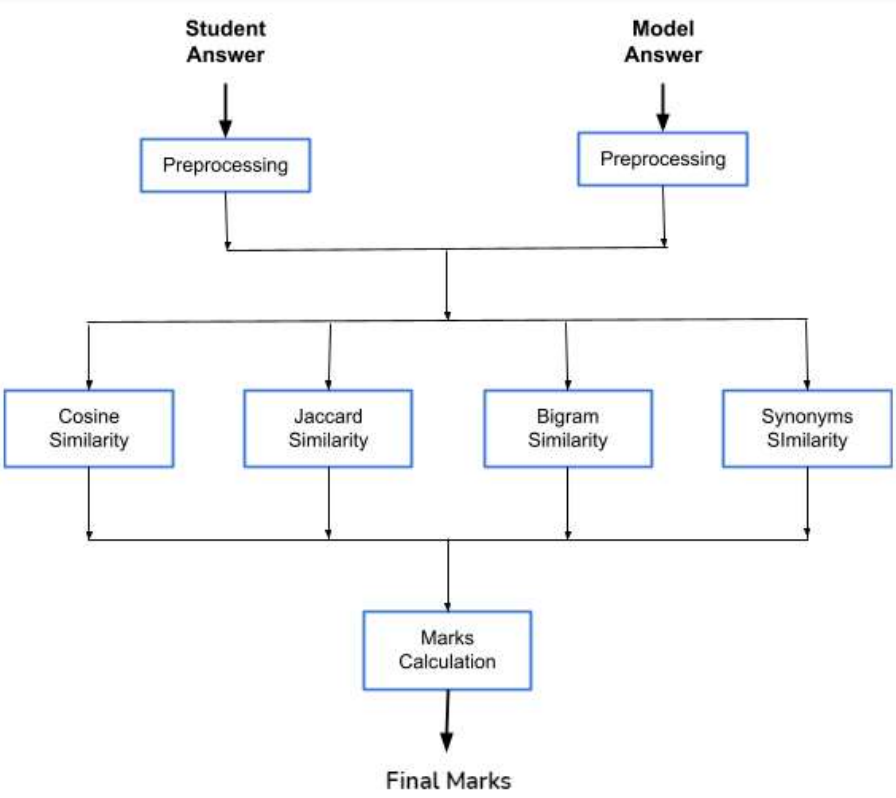

Figure 3: Flow of process in answer evaluation using similarity measures

The next step is to check the extent to which these lists are similar. The overall similarity score is calculated using different types of similarities, namely cosine similarity, jaccard similarity, synonyms similarity and bigram similarity.

i**) Cosine Similarity:** Cosine Similarity is the similarity between the vector representation of two words in n-dimensional space. Cosine similarity gives accurate similarity even if two texts are of different sizes. Equation 5 is used to calculate cosine similarity.

$$Cosine\ Similarity = \frac{A.B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i\ B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad \text{..... (5)}$$

ii) Jaccard similarity: Jaccard Similarity coefficient measures the similarity and diversity between two texts. The formula for calculating jaccard similarity is shown in equation 6.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad \text{..... (6)}$$

Where A and B are the list of words obtained after preprocessing the understudy answer and benchmark answer. The intersection defines how many words both lists have in common and the union is the combination of words in both lists.

iii) Bigram-Similarity: The bigram similarity checks the similarity between contiguous sequences of size two. Bigram similarity can be applied to any language as it is language independent.

iv) Synonyms Similarity: This approach also uses the synonyms similarity to ensure that the understudy answers that fail to match the model answer get a rational score if they are equivalent to the model answer. Basically, the synonym of the understudy answer is found if it does not resemble the model answer. This is done by finding synonyms of the stopwords present in the answers.

C. Assign Weight:

Since we are using four different similarity measures, it is required to assign weight to these measures as needed. These weights are calculated experimentally by manually checking the answers and noticing the weight of these measures in the final marks. These experiments have been performed on 50 sets to improve the exactness of these weights. The final weights are derived using the research of Rahman et al.[31]. The final weights obtained are as follows.

- ✓ Synonyms similarity: 0.45
- ✓ Bigram Similarity: 0.37
- ✓ Jaccard Similarity: 0.09
- ✓ Cosine Similarity: 0.09

D. Marks Evaluation:

After getting the above mentioned similarity scores, the final mark is calculated using equation 7.

$$Total_{marks} = 0.09 * J + 0.37 * B + 0.09 * C + 0.45 * S \quad \text{..(7)}$$

Where J is Jaccard Similarity, B is Bigram Similarity, C is Cosine Similarity and S is synonyms Similarity

3.3 Answer Evaluation using sequential model

With the arrival of new recurrent neural network architectures like Long Short Term Memory(LSTM), the task of Automated Text Scoring(ATS) has become easier. In the light of this approach, we use a sequence-to-vector model with 2 LSTM layers for scoring essay-type answers. A long answer/essay and a keyword set is input to this module which outputs the score to the essay. The process flow for the ATS using deep learning is shown in figure.4.

The Kaggle's ASAP-AES (Automated Student Assessment Prize - Automated Essay Scoring) by the Hewlett foundation was used for training the sequence to vector model. The dataset consists of essays written by students and the scores provided by one or more human experts. The dataset involves 8-essay sets (length 150-550 words) per response. There are about 12977 students' responses on the essay sets. All the essays are marked out of maximum marks based on their length and set value. Our model is trained to score a long answer on the basis of length. The NERs in the essay have been replaced by corresponding tags to remove personally identifying information. For e.g., "I attend Springfield School..." becomes "I attend @ORGANIZATION1...".

A. Input Description

Input consists of the student's answer and expected keywords that must be included in the answer. The keywords are included to test the relevance of answers with the context, as the model trained on ASAP-AES dataset scores anonymized essays.

B. Data Preprocessing

Since the sequential models work on vectors, we need to preprocess the data before using it for training, testing or evaluation. First of all, the entire data set is divided into training and test dataset. The training dataset is further split into a training set and a validation set.

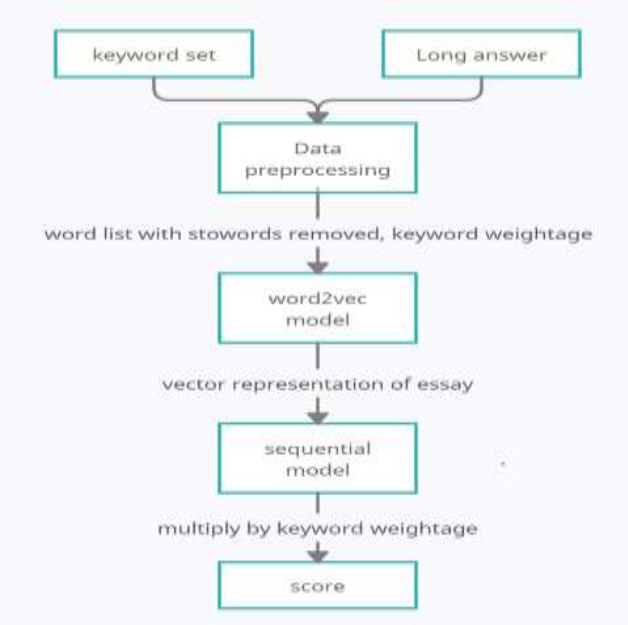

Figure 4: Methodology for ATS using deep learning

Steps involved for preprocessing the data for the sequential model are sentence tokenization, word tokenization, removal of stopwords. The remaining words are used as training data for the word2vec model.

C. Model training and validation

Our deep learning approach for ATS uses two models for obtaining vector representation (word2vec) and for scoring the answers (sequential), both of which are elaborated in this section.

i) Word2Vec model

Word2Vec in general, is an algorithm used for producing distributed representation of words. Our system includes training of the word2vec model from the gensim library. The model learns vocabulary based on tokens contained in the text feeded to it. The feature vector for student's answer is obtained as shown below:

- Count the number of words in the student answer that were present in the word2vec vocabulary
- Assign a vector corresponding to the word in a vector of desired dimensions which represents the sentence embedding
- Normalize the array by dividing the entire vector obtained in step 2 by the output of step 1
- In order to obtain the vector representation of an essay, n-dimensional vector representation for each of the sentences is pushed into a vector to obtain a L x n vector, where L denotes the number of sentences in the essay and n denotes the dimension of the sentence embedding used.

ii) Sequential model

The task of ATS requires a model that can associate a sequence (or essay) with vectors (or scores). Hence the Sequential model (keras) is trained for scoring the essay-type answers. The model consists of several layers, the most important one being the LSTM layers. Long short term memory (or LSTM) is a special kind of Recurrent Neural Network (RNN). It is special due to its capability to retain long-term dependency learning.

Our sequential model uses stacked LSTMs since stacked models give better results as stated by many researchers. Furthermore, it is trained and validated on k-folds of the data to increase the learning through perpetual calculation of loss in each round. The entire sequential model consists of following layers:

- LSTM sequence to sequence layer: This is the input layer to which the vector representation of the essays is feeded one at a time. It returns a sequence of vectors representing the hidden state for each input time-step, the hidden state output and the cell state for the last input time step.
- LSTM sequence to vector layer: It is feeded with the output generated at the first layer of LSTM. Addition of this layer is done for extracting more abstract information. It returns a single vector representing one hidden state for each input.

- Dropout layer: This layer is added to prevent the model from overfitting. During training, this layer sets random inputs to 0 with a pre-specified frequency rate.
- Dense layer: It is a regular fully-connected neural network layer. This layer was added in the model to provide learning features from all combinations. It is stacked after LSTM because many frameworks give the internal (or hidden) state h as output. The dimensionality of this output (which is the number of units) may not be equal to the dimensionality of the target that we want. Dense layer allows us to tune the dimension of output.

The model is feeded with uniformly-dimensioned vectors representing the essays and predicts the score for the answer. The length of intersection of the keyword set input by the examiner and the student's answer is considered to have weightage(on a scale of 0 to 1) over the marks obtained. Consequently, we multiply this weightage to the score predicted by the sequential model

IV. EVALUATION

The results of our first method (answer evaluation system using keywords matching) is compared with the marks allotted manually by teachers answers taken from ASAP-SAS dataset[21] . Results of 10 sample are shown in table 1, the second column shows the marks allotted in manual checking (from dataset) and the third column shows the marks allotted by our system. The marks obtained are close and the results of our system are satisfactory.

Sample No.	Marks allotted in manual checking	Marks allotted by our system	Squared Error
1	2.5	2.9	0.16
2	3.22	3.5	0.078
3	4.00	3.42	0.3364
4	2.21	2.26	0.0025
5	3.5	3.18	0.1024
6	4.2	3.4	0.64
7	3.5	3	0.25
8	4.5	4	0.25
9	4.5	3.9	0.36
10	3.7	4.1	0.16

Table 1: Comparison between manual checking and by the keyword matching evaluation system

The performance of the first method is measured using root mean squared error value using Root Mean Squared Error as shown in equation 8 .

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \quad \text{.....(8)}$$

In result analysis of 2000 samples RSME value is 0.252012678 for method 1.

Same evaluation process as discussed for the first method is used to check the performance of our second method. Our second system is also giving satisfactory results as shown in table 2.

Sample No.	Marks allotted in manual checking	Marks allotted by our system	Squared Error
1	2.96	3.70	0.5476
2	3.22	2.98	0.0576
3	4.00	3.84	0.0256
4	2.21	2.28	0.0049
5	3.5	3.58	0.0064
6	4.2	4.78	0.3364
7	3.9	3.22	0.4624
8	4.5	4.22	0.0784
9	4.5	3.79	0.5041
10	3.7	3.45	0.0625

Table 2 :Comparison between manual checking and by the similarity measure evaluation system.

In result analysis of 2000 samples RSME value is 456716542 for method 2.

The approach 3 for answer evaluation includes usage of a deep learning model which was trained on ASAP-AES. The answer evaluation model gives a quadratic(or Cohen's) kappa score of 0.96 upon being tested on 25% of the ASAP-AES dataset. Cohen's Kappa (shown in equation 9) measures the agreement between two raters who classify N items into C mutually exclusive categories.

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e} \quad \text{.....(9)}$$

where, po is the relative observed agreement among raters, and pe is the hypothetical probability of chance agreement. Kappa score is 1 if both rates are in complete agreement.

A scatter plot of the predicted scores over the actual scores in the test dataset is shown in Figure. 5, where the x-axis represents various essays(or student answers) and y-axis represents the scores obtained.

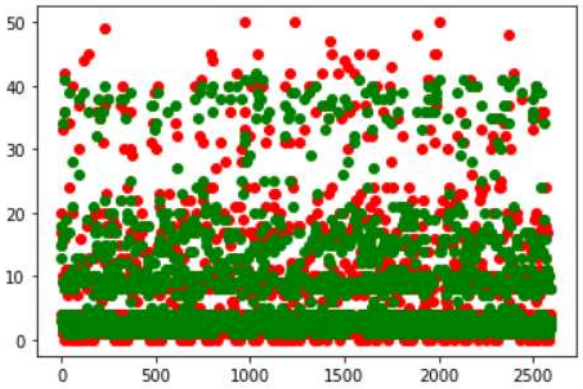

Figure 5: Scatter plot of test scores(red) vs. predicted scores(green)

V. RESULT

In this research three different systems evaluate the answers successfully. The handwritten text extraction works fine for good handwriting but may falter for bad ones. Also, the output of a handwritten text extraction system may contain the altered sequences of words in the image. Such a system could only be supported in the keyword matching approach, which can compare the tokens in extracted text with the keyword list for scoring the answers. The results obtained from each are shown below:

Answer evaluation using keyword matching:

The system marks satisfactorily form the domains where presence of keywords plays an important role. A sample output generated when an answer derived from wikipedia article [23] is input to the system, is shown below:

Input: keywords= ['symbol', 'table', 'compiler', 'processor', 'object', 'interpreter', 'execute', 'lexical', 'symantic', 'syntactic', 'analysis', 'parser' ,'decoder', 'high', 'level', 'language', 'token', 'low', 'source', 'machine', 'assembly', 'program', 'code', 'translates']

expected_no_of_words = 100

expected_no_of_sentences = 5

maximum_marks = 10

expected_keywords = 12

answer = '"A compiler is a special program that processes statements written in a particular programming language and turns them into machine language or "code" that a computer's processor uses. Typically, a programmer writes language statements in a language such as Pascal or C one line at a time using an editor. The object code is machine code that the processor can execute one instruction at a time.'"

Output:

Matching Keywords= [Object, compiler, program, processor, machine, execute, code, programmer, Language]

keywords_percentage = 0.48750000000000004

word_percentage = 0.1825

sentence_percentage = 0.06

Total Marks: 7.3 / 10

Answer evaluation using similarity measures:

The system performs well with short answers by measuring different similarities (cosine similarity, jaccard similarity ,bigram similarity and synonyms similarity) between the student answer and model answer. A sample output generated using derivations from wikipedia articles as model answer[24] and student answer[25] is shown below using the :

Input: Model Answer about C Programming language:

'The C programming language is a structure oriented programming language, developed at Bell Laboratories in 1972 by Dennis Ritchie C programming language features were derived from an earlier language called "B" (Basic Combined Programming Language —BCPL) C language was invented for implementing the UNIX operating system. In 1978, Dennis Ritchie and Brian Kernighan published the first edition "The C Programming Language" and commonly known as K&R C In 1983, the American National Standards Institute (ANSI) established a committee to provide a modern, comprehensive definition of C. The resulting definition, the ANSI standard or "ANSI C", was completed late 1988'.

Student Answer:

In 1972 Dennis Ritchie at Bell Labs writes C and in 1978 the publication of The C Programming Language by Kernighan & Ritchie caused a revolution in the computing world In 1983, the American National Standards Institute (ANSI) established a committee to provide a modern, comprehensive definition of C. The resulting definition, the ANSI standard, or "ANSI C", was completed late 1988.

Output:

Jaccard Similarity: 0.4098360655737705

Bigram Similarity: 0.38016528925619836

Cosine Similarity: 0.6746395048753531

Synonyms Similarity: 0.42276422764227645

Total marks = 8.770583241857603/10

Answer evaluation using deep learning approach

The approach 3 for answer evaluation included usage of a deep learning model which was trained on ASAP-AES. This module successfully evaluates the student's answers in text format, a sample of which is shown below. The answer1 and answer 2 are derived from wikipedia articles [26] and [27] respectively.

Input Keywords: ['Apple', 'company', 'technology', 'Steve', 'Jobs', 'America', 'electronics', 'software']

Answer 1: '"Apple Inc. is an American multinational technology company headquartered in Cupertino, California, that designs, develops and sells consumer electronics, computer software, and online services. It is considered one of the Big Tech technology companies, alongside Amazon, Google, Microsoft, and Facebook. The company's hardware products include the iPhone smartphone, the iPad tablet computer, the Mac personal computer, the iPod portable media player, the Apple Watch smartwatch, the Apple TV digital media player, the AirPods wireless earbuds and the HomePod smart speaker. Apple's software includes macOS, iOS, iPadOS, watchOS, and tvOS operating systems, the iTunes media player, the Safari web browser, the Shazam music identifier and the iLife and iWork creativity and productivity suites, as well as professional applications like Final Cut Pro, Logic Pro, and Xcode. '"

Answer 2: '"Elon Reeve Musk was born in 1971. He is a business magnate, industrial designer, and engineer. He is the founder, CEO, CTO, and chief designer of SpaceX, early investor, CEO, and product architect of Tesla, Inc. He is also the founder of The Boring Company; co-founder of Neuralink; and co-founder and initial co-chairman of OpenAI. A centibillionaire, Musk is one of the richest people in the world. Musk was born to a Canadian mother and South African father. He was raised in Pretoria, South Africa. He briefly attended the University of Pretoria. He later moved to Canada aged 17. He then attended Queen's University. He transferred to the University of Pennsylvania two years later. He received

dual bachelor's degrees in economics and physics there. He moved to California in 1995 for Stanford University. But he decided instead to pursue a business career. He then co-founded the web software company Zip2 with his brother Kimbal. The startup was acquired by Compaq for $307 million in 1999. Musk co-founded online bank X.com that same year. It merged with Confinity in 2000. This later formed the company PayPal, PayPal was subsequently bought by eBay in 2002 for $1.5 billion.'"

Output

The system provides a score of 8.5 to the first answer, while a score of 2.5 to the second answer for the given keyword list, which is quite close to what we want to achieve.

VI. CONCLUSION AND FUTURE WORK

Our system successfully evaluates answers from different domains using the three mentioned approaches. As the approaches use different courses of action, the application of our system is expanded. The system can be improved by using a more efficient technique of handwritten text extraction (for bad handwritings) and employing it to the entire system. Another functionality that can be added is grammar and spelling check to deduce marks over grammatical mistakes, which is quite an important feature for a subject of language domain.

REFERENCES

[1] M. D. Shermis & J. Burstein (Eds.). Page, E. B. (2003). Project Essay Grade: PEG. In, Automated essay scoring: A cross-disciplinary perspective. Lawrence Erlbaum Associates Publishers, pages 43-54.

[2] Piyush Patil, Sachin Patil, Vaibhav Miniyar and Amol Bandal. 2018. Subjective Answer Evaluation Using Machine Learning. International Journal of Pure and Applied Mathematics, volume 118.

[3] Thomas K. Landauer, Darrell Laham, and Peter W. Foltz. 2003. Automated scoring and annotation of essays with the Intelligent Essay Assessor. In M.D. Shermis and J.C. Burstein, editors, Automated essay scoring: A cross-disciplinary perspective, pages 87–112.

[4] L.M. Rudner and Tahung Liang. 2002. Automated essay scoring using Bayes' theorem. The Journal of Technology, Learning and Assessment, 1(2):3–21.

[5] Keisuke Sakaguchi, Michael Heilman, and Nitin Madnani. 2015. Effective feature integration for automated short answer scoring. Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications.

[6] S. Impedovo, L. Ottaviano and S.Occhinegro, " Optical character recognition ," International Journal Pattern Recognition and Artificial Intelligence, Vol. 5(1-2), pp. 1- 24, 1991

[7] George Nagy, Stephen V. Rice, and Thomas A. Nartker, " Optical Character Recognition: An Illustrated Guide to the Frontier (The Springer International Series in Engineering and Computer Science)

[8] J.Pradeep , E.Srinivasan and S.Himavathi, " Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Networks ", International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011.

[9] B vanni, M. shyni, and R. Deepalakshmi, "High accuracy optical character recognition algorithms using learning array of ANN" in Proc. 2014 IEEE International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2014 International Conference.

[10] Baoguang Shi, Xiang Bai and Cong Yao. 2015. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. arXiv:1507.05717v1 [cs.CV].

[11] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. Feb.

[12] Duyu Tang. 2015. Sentiment-specific representation learning for document-level sentiment analysis. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM '15. Association for Computing Machinery (ACM).

[13] Dimitrios Alikaniotis, Helen Yannakoudakis and Marek Rei. 2016. Automatic Text Scoring Using Neural Networks. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 715–725,

[14] V. Lakshmi and Dr. V. Ramesh. 2017. Evaluating Students' Descriptive Answers Using Natural Language Processing and Artificial Neural Networks. International Journal of Creative Research Thoughts (IJCRT1704424), volume 5.

[15] Y. Oganian, M. Conrad, A. Aryani, K. Spalek and H. R. Heekeren, "Activation Patterns throughout the Word Processing Network of L1-dominant Bilinguals Reflect Language Similarity and Language Decisions," in Journal of Cognitive Neuroscience, vol. 27, no. 11, pp. 2197-2214, Nov. 2015.

[16] Mark D Shermis. 2015. Contrasting state-of-the-art in the machine scoring of short-form constructed responses. Educational Assessment, volume 20(1), pages 46–65.

[17] A. R. Lahitani, A. E. Permanasari and N. A. Setiawan, "Cosine similarity to determine similarity measure: Study case in online essay assessment," 2016 4th International Conference on Cyber and IT Service Management, Bandung, 2016, pp. 1-6.

[18] T. Bluche, C. Kermorvant, C. Touzet and H. Glotin, "Cortical-Inspired Open-Bigram Representation for Handwritten Word Recognition," 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, 2017, pp. 73-78.

[19] J. Nau, A. H. Filho and G. Passero, "Evaluating Semantic Analysis Methods for Short Answer Grading Using Linear Regression, " PEOPLE: International Journal of Social Sciences (2017), Volume 3 Issue 2, pp. 437 – 450.
[20] Google Vision API, https://cloud.google.com/vision/
[21] The Automated Student Assessment Prize - Short Answer Scoring (ASAP-SAS). 2012. Sponsored by The Hewlett foundation in Kaggle contest: https://www.kaggle.com/c/asap-sas
[22] The Automated Student Assessment Prize - Automated Essay Scoring (ASAP-AES). 2012. Sponsored by The Hewlett foundation in Kaggle contest: https://www.kaggle.com/c/asap-aes
[23] Wikipedia contributors. (2021, June 11). Compiler. In Wikipedia, The Free Encyclopedia. Retrieved 09:37, June 14, 2021, from https://en.wikipedia.org/w/index.php?title=Compiler&oldid=1028004430
[24] Wikipedia contributors. (2021, June 1). C (programming language). In Wikipedia, The Free Encyclopedia. Retrieved 09:38, June 14, 2021, from https://en.wikipedia.org/w/index.php?title=C_(programming_language)&oldid=1026383874
[25] Wikipedia contributors. (2021, June 1). C (programming language). In Wikipedia, The Free Encyclopedia. Retrieved 09:42, June 14, 2021, from https://en.wikipedia.org/w/index.php?title=C_(programming_language)&oldid=1026383874
[26] Wikipedia contributors. (2021, June 9). Apple Inc.. In Wikipedia, The Free Encyclopedia. Retrieved 09:44, June 14, 2021, from https://en.wikipedia.org/w/index.php?title=Apple_Inc.&oldid=1027706334
[27] Wikipedia contributors. (2021, June 13). Elon Musk. In Wikipedia, The Free Encyclopedia. Retrieved 09:45, June 14, 2021, from https://en.wikipedia.org/w/index.php?title=Elon_Musk&oldid=1028429145
[28] B. S. J. Kapoor, S. M. Nagpure, S. S. Kolhatkar, P. G. Chanore, M. M. Vishwakarma and R. B. Kokate, "An Analysis of Automated Answer Evaluation Systems based on Machine Learning," *2020 International Conference on Inventive Computation Technologies (ICICT)*, 2020, pp. 439-443, doi: 10.1109/ICICT48043.2020.9112429.
[29] Mahmud, Tamim & Hussain, Md Gulzar & Kabir, Sumaiya & Ahmad, Hasnain & Sobhan, Mahmudus. (2020). A Keyword Based Technique to Evaluate Broad Question Answer Script. 10.13140/RG.2.2.20912.92164.
[30] Bharadia, Sharad & Sinha, Prince & Kaul, Ayush. (2018). Answer Evaluation Using Machine Learning.
[31] Md. Motiur Rahman, Ferdusee Akter. "An Automated Approach for Answer Script Evaluation Using Natural Language Processing". CSET 2019,pp. 39-47.