RESEARCHERID
THOMSON REUTERS

[ICRTCET-2018]

ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

A STUDY ON AUTOMATIC DESCRIPTIVE ANSWER SCRIPTS EVALUATOR

Parkavi A[*1], Bhavana G Dongre[2], Gayathri Devi G[3], Spoorthy S. Hathwar[4] & PatnamPranitha[5]
[*1,2,3,4&5]Ramaiah Institute of Technology

ABSTRACT

In educational institutions a major concern is evaluation of answer sheets of both objective and subjective type questions. Automatic Descriptive Answer Scripts Evaluator aims at making this time consuming tedious task of correcting students answers scripts very easy. For that the study carried out in this paper is spread over the techniques like text matching based on Jaccard similarity, Dice's co-efficient, cosine similarity and semantic similarity.

Keywords: *Cyber Security, Artificial Intelligence, Intrusion detection.*

I. INTRODUCTION

Automatic Descriptive Answer Scripts Evaluator helps in automatically evaluating the answer scripts based on the reference answer key that it is provided with. There are tools to evaluate multiple choice questions but evaluation of descriptive answers is more complex and not readily available. There are several advantages of using this automated system like it helps in reducing the time taken by faculty to correct the papers, the tests or examinations can be conducted online, and the answers can be evaluated immediately and it would be beneficial for universities, schools and colleges for academic purpose by providing ease to faculties and the examination evaluation cell. Some algorithms are used to match the answers with that of the reference key. Algorithms used to implement text matching are Jaccard similarity, Dice's co-efficient, Cosine Similarity and Semantic Similarity.

A. Jaccard similarity: Jaccard index measures the similarity between two documents based on the intersection and union of two sample documents.

Where A and B are 2 documents to checked for similarity

B. Dice's coefficient: Dice's coefficient is similar to Jaccard index but with double weights.

where A and B are 2 documents to be checked for similarity.

C. Cosine Similarity: Cosine similarity is a measure of similarity between two nonzero vectors of an inner product space that measures the cosine of the angle between them. In text analysis, each vector can represent a document. The greater the value of θ, the less the value of cos θ, thus the less the similarity between two documents.

D. Semantic Similarity: Semantic similarity is a metric defined over a set of documents or terms, where the idea of distance between them is based on the likeness of their meaning or semantic content i.e. it measures the distance of meaning of two terms.

E. Levenshtein Similarity: Levenshtein Distance is the minimum edit distance which is used to find the minimum edits required to make two strings similar. If minimum edits required, maximum will be the similarity.

II. RELATED WORK AND LITERATURE SURVEY

There are a number of papers on automatic evaluation of descriptive answers. Shweta M. Patil has developed a system which is based on computer assisted assessment that uses NLP technique. The project is divided into different modules and each module performs a specific functionality. Techniques used for matching answers are POS tagging where parts of speech are identified and synonyms and antonyms of the keywords are identified.

Scores are given if the collective meaning is similar to the answer key and scores are reduced if antonyms of the keywords are present [4]. Praveen S has developed a system that uses techniques like pre-processing, POS tagging, Parse tree construction, knowledge representation and inference system [5]. He has used Wordorder similarity measure which is used to find the similarity between the texts. Similarity matrix is built to compare the similarity between the texts. Based on the score of the similarity matrix, grades are awarded. The more the similar, higher is the score[1]. In another research work by K. Meena, evaluation of student's answers is done using Hyperspace Analogue to language procedure and Self organizing map. HAL is a numeric method for analysing text. It does so by running a sliding window of fixed length across a text, calculating a matrix of word co-occurrence. Self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples

The approach followed by another researcher [2] includes deriving key concepts and their description from the model answer as well as from candidate answer. Candidate's key concepts and description are matched against those of model answer and there by evaluation is carried out [3]. This paper splits the reference answer and the student answer into Bag of Words(BOW) and computes similarity using Syntactic Features – using n-grams, Basic Similarity Features – minimum, maximum, median, Spelling Features

Entailment Features, Semantic similarity features, Synonym Consideration [9]. In this research work,the researcher has been finding of similarity between reference answer and student answer using the methods of Uni Gram Ref Student, Unigram Student Ref, Lemma Ref Student, Lemma Student Ref, Bigram Ref Student. Bigram StudentRef [8]. In a research work byPranaliNikam, Mayuri Shinde, the student answer and the reference answer are converted into a graphical representation and thesimilarity between the two are computed using similarity measures like String MatchPartial string match, WordNet, Full string match. In this research work, Synonyms of words are considered, but grammatical errors are neglected [6].

Another system is developed, which uses the technique of matching student answer with reference answer by converting them intographical form and matching their nodes and labels. It considers syntactic components likeNoun Phrase, Verb Phrase etc. Similarity measures are String Match – abbreviations,case change, morphological change, tensechange etc. WordNet uses metrics like is_similar_to, is_apart_of, is_oppositeetc. and similarity index is obtained [7].

Saha, et al., propsed Intelligent Tutoring System that teaches students and evaluates their performances and provides a proper feedback. It checks on the answers with appropriate spelling and grammar under restriction. It works on respected answer given and distinguishes them whether correct , error or elaboration. The marks for an answer (composed of simple sentences) is given based on the keywords, synonyms and antonyms, the weights of which are different [10]. Qureshi in his research work proposed an electronic system that generates question papers automatically and automatic subjective answer assesment. The system generates the question paper randomly for each individual student based on the course learning outcomes and complexity like simple, average and difficult equally distributed . The subjective assessment system automatically assesses the answer to match the keywords using a knowledgebase and data mining techniques to grade a student [11].

Tetali, et al., in their research work developed a tool TadaCo to automatically evaluate descriptive answers and assessment of corresponding course outcomes. TadaCo works either in a Semi-automated mode or in Complete automated mode. Semi-automated mode has the flexibility to allow the faculty to reevaluate an answer and update the results. It is established that the Semi-automated mode yields comparatively better results than the Complete automated mode. It evaluates the descriptive answers by matching keywords and phrases in the answer given by the disciple, with the keywords and phrases of the original answer [12].Mohan, et al., proposed CosInfo algorithm to evaluate the papers automatically. This algorithm implemented the feature clustering for evaluation purpose that calculated the similarity between two documents and cluster the relevant documents into different groups. Proposed algorithm used the expected information function and parts of speech in English grammar as parameters to cluster the data (cosine similarity formula), and also build a model to classify the testing documents using SVM

classification to assess the degree of similarity which will help to award the marks automatically. Experimental results showed that the proposed method obtains better and accurate results to allocate marks compared with manual evaluation [13].

Keiji Yasuda has proposed a system that gives score to descriptive answers automatically. Methods used are long short-term memory recurrent neural networks. Words in the sentence are classified according to the formulas; LSTM calculates forget gate, input gate parameters for the sentence. By using the formula for L1 norm, similarity between the students answer and model answer. Average score of model answers is calculated and assigned scores to student answers [14].

Hyo-Jung Oh, et al., have proposed a system generates automatic answer from encyclopaedia for the proposed question. Here DAT type answers are given by identifying the patterns in the question and matching with the DIU pattern matching is based on finite automata [15].

RuhiDubey, et al., used inbuilt classification algorithms of weka tool are to evaluate the answers automatically. Data regarding the questions and the model answers are collected and scoring is given manually. Tokenizer breaks sentence into words based on punctuation. Vectorization is done with tokens obtained and a class is assigned to each vector. Training is done by using random forest and FT classifier, raptree classifier. Filtering is done to make test and training data size same. Classification is done accordingly [16].

Different evaluation designs are explained in this paper. Some of them are Random assignment, use of control variables, natural variation and the quasi experimental design. Correlation design is used for the data which can be connected by some similarities [17].

III. PROBLEM DEFINITION

There are several tools for online exam evaluation but most of them are for multiple choice questions. Our paper aims to evaluate the answers for subjective questions. This helps in analysing the student's performance at higher level of taxonomy of educational objective. This system finds the collective meaning of the answer and compares with the answer key and awards the marks based on the similarity extent.

IV. PROPOSED SYSTEM

The proposed system of ours, uses different modules and techniques to increase the accuracy and efficiency as shown in Fig. 1.We have designed our system for evaluation of answer scripts using different techniques. And then we have compared the output of the evaluation of different techniques.

The modules used are:

- ✓ Students' answer intake

This module takes the answers given by the student for the particular question.

- ✓ Teacher's solution intake

This module takes the reference answer entered by the tutor. It must contain all the keywords for that particular answer.

- ✓ Similarity check

The two solutions are checked against each other using Jaccard similarity, Dice's coefficient, cosine similarity and semantic similarity.

- ✓ Grading system

Based on the results of the similarity check, the grades are awarded accordingly. The greater the similarity, higher is the score.

ISSN 2348 – 8034
Impact Factor- 5.070

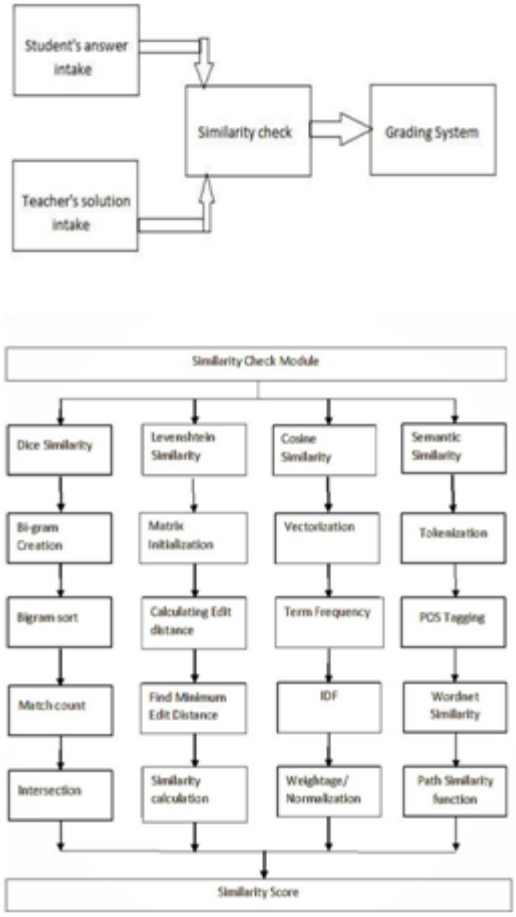

Fig. i. Automated student answer script evaluator

V. IMPLEMENTATION

The first phase of implementation involves pre-processing techniques like stop-word removal, stemming, etc. to remove unwanted words or phrases. For determiningthe similarity, we have used the techniques namely: Jaccard Similarity, Dice's coefficient, Cosine similarity and Semantic similarity.

Jaccard index measures the similarity between two documents based on the intersection and union of two sample documents. Thistechnique uses intersection - union formula to calculate the similarity between student answer sheet and the sample answer sheet. It parses each word in the students answer sheet and sample papers also. It takes the intersection of set of words in student's answers and the sample answer sheets. The common words in both the documents will be found out by using intersection of two documents. To give the score and normalize the count of the common elements the same is divided by the union of words in the two documents. In this way the students answer similarity with sample papers is determined and the J(A,B) gives the score for answer sheet of each student when compared with the sample documents and answer sheet.

Dice's coefficient is similar to Jaccard index but with double weights. Before applying Dice's coefficients, the text in the document is divided into substrings of length 'n' (n -grams). Each substring obtained from the n-gram is changed into corresponding hash value. This is done by using MD5. Then the most occurring hash value is selected using a prime number 'p'. The frequently occurring hash value will be given by 'hash_value mod p'. This is used for the similarity check using Dice's co-efficient. The common words in both the documents will be found out by using

intersection of two documents. To give the score and normalize the count of the common elements the same is divided by the addition of words in the two documents. In this way the students answer similarity with sample papers is determined and the s(A,B) gives the score for answer sheet of each student when compared with the sample documents and answer sheet.

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. In text analysis, each vector can represent a document. The greater the value of θ, the less the value of cosθ, thus the less the similarity between two documents. Term Frequency (TF) and Inverse Document Frequency (IDF) are used in computing Cosine Similarity.

Term Frequency measures the number of times a word occurs in a document. Since each document will be of a different size, the Term Frequency (TF) need to be normalized. Finding relevant documents is done using Inverse Document Frequency (IDF). All terms in a document might not have equal weightage. So, we need to weigh up or weigh down the effect of terms occurring frequently or rarely. Logarithms are used to determine these. Thus, greater the cosine value, higher is the similarity between the two texts.

Semantic similarity is a metric defined over a set of documents or terms, where the idea of distance between them is based on the likeness of their meaning or semantic content i.e. it measures the distance of meaning of two terms. In semantic similarity, POS tagging, WordNet, Natural language processing techniques are used to find the meaning of the two solutions.

POS tagging is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc. WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members which is used to compare the similarity between two texts. These techniques give a similarity score which directly indicates the percentage of similarity between the two sets of solutions and hence the score to be awarded.

Levenshtein Distance is the minimum edit distance which is used to find the minimum edits required to make two strings similar. The strings are taken in two dimensional matrix forms. Each substring of the strings is taken and the number of edits required is calculated for each substring of two strings and populated in the matrix.

Each entry in the matrix is based on the submatrix of the matrix. Dynamic programming techniques are used to calculate the levenshtein distance for each substring pair. Numpy package of python is used for the implementation. Last entry gives the number of edits required to convert one string to other. Similarity of two strings is inversly proportional to the number of edits. Similarity percentage is calculated by using the formula
Similarity = (length(longest string) – number of edits)/ (length(longest string))

VI. RESULTS AND DISCUSSION

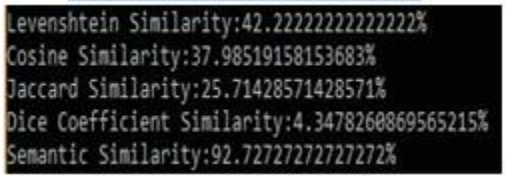

Fig. ii

Fig ii is the results' snapshot after running all the four algorithms for a particular question and answer pair. From the results we can conclude that the accuracy of the algorithms are in the following order:

Semantic >Levenshtein> Cosine >Jaccard> Dice

VII. CONCLUSION

Thissystem helps in automating the task of correcting the answer scripts by using similarity measures to check the similarity between the two texts and grade them accordingly based on different similarity checking techniques. It can be used in schools, colleges to reduce the burden on teachers and also to make the process faster. Its can also be used in various online entrance exams.

VIII. FUTURE WORK

The accuracy level can be further improved by using other NLP techniques and using a training model. The present system only gives a score or percentage, it can be further extended to give feedback of the student's performance and strengths and weaknesses of the student.

REFERENCES

1. *Naser S Al Madi, Javed I Khan (2017): A comprehension-based framework for measuring semantic similarity*
2. *Yassine Mrabet, Halil Kilicoglu, Dina Demner-Fushman (2017): TextFlow- A Text Similarity Measure based on Continuous Sequences*
3. *Madhumitha Ramamurthy, Ilango Krishnamurthi (2016): Design and Development of a Framework for an Automatic Answer Evaluation System Based on Similarity Measures*
4. *Chenguang Wang, Yangqiu Song, Haoran Li (2015) : KnowSim- A Document Similarity Measure on Structured Heterogeneous Information Networks*
5. *Vani K, Deepa Gupta (2015): Investigating the impact of combined similarity metrics and POS tagging in extrinsic text plagiarism detection system*
6. *Answer evaluation of short descriptive answers. (n.d.).*
7. *K. Meena, L. R. (2014). Evaluation of the descriptive type answers using hyperspace analog to language and self-organizing map.*
8. *Misha D. Chandana, A. V. (n.d.). Descriptive Answer Evaluator.*
9. *Ms.Shweta M. Patil, P. M. (2014). evaluating student descriptive answer using nlp. International Journal of Engineering Research & Technology*
10. *(IJERT), 4.*
11. *Praveen, S. (2014). An Approach to Evaluate Subjective. International Journal of Innovative Research in Computer, 4.*
12. *Automatic Evaluation of Descriptive Answer using Pattern Matching Algorithm PranaliNikam, MayuriShinde, Rajashree Mahajan, ShashikalaKadam*
13. *Algorithm for Automatic Evaluation of Single Sentence Descriptive Answer AmarjeethKaur, MSasiKumar, ShikaNema, Sanjay Pawar*
14. *CU: Computational Assessment of Short Free Text Answers – A tool for evaluating student's understanding IfeyinwaOkoye, Steven Bethard, Tamara Sumner*
15. *UKP – BIU: Similarity and Entailment Metrics for Student Response Analysis TorstenZesch, OmerLevy, Iryna, Ido Dagan*
16. *M. Saha and M. Chakraborty: A Novel Approach for Descriptive Answer Script Evaluation*
17. *M. R. Qureshi: A Proposal Of Electronic Examination System To Evaluate Descriptive Answers*
18. *D. R. Tetali, S. Tamma and L. Ramana: Tool for Automated Evaluation of Descriptive Answers & Course Outcomes (TADACO)*
19. *A. K. Mohan and M. K. Prasad: A Novel Feature Clustering Algorithm for Evaluation of Descriptive Type Examination*
20. *Keiji Yasuda: Automatic Scoring Method for Descriptive Test Using Recurrent Neural Network*
21. *Victor U. Thompson, Christo Panchev, Michael Oakes (2015): Performance evaluation of similarity measures on similar and dissimilar text retrieval*
22. *Eko Sakti Pramukantoro, M. Ali Fauzi (2016): Comparative analysis of string similarity and corpus-based similarity for automatic essay scoring system on e-learning gamification*

23. Feddy Setio Pribadi, Teguh Bharata Adji, Adhistya Erna Permanasari, Anggraini Mulwinda, and Aryo Baskoro Utomo (2017): Automatic short answer scoring using words overlapping methods