Benevolent Fuzzy Text Response Evaluator

[1]Aritra Saha, [2]Somerita Banerjee
[1]Assistant Professor, [2]Assistant Teacher
[1]Heritage Institute of Technology – Kolkata
[2]Delhi Public School, New Town

Abstract: --In this digital age, aspects of learning and evaluating have also taken the digital route. Various techniques exist for evaluating the response from a learner, most common among them being MCQ (multiple choice questions) using OMR (Optical Mark Reader) sheets for a rapid turnaround. But there have been several arguments against such testing systems, citing reasons such as they lack the ability for testing complete knowledge, or that they can be beaten with some intelligent guess work. Bottom line of all these arguments being the lack of an expert human evaluator who can use the discretionary powers of the trained mind to weed out inadvertent mistakes of the learner while being able to identify the lack of knowledge.This paper aims to propose an effective scheme for smart identification of learners' text-based response in an e-Learning environment. In this paper, we present an intelligently adaptive mechanism which accepts single-word textual response typed-in by the learner during their interaction with the e-learning system. The system then attempts to mimic the intelligence and discretion of a benevolent human evaluator who understands the difference between lack of knowledge and unintentional mistakes that might be committed by the learner while responding to the system's queries. We want to model an automated text response evaluator that can emulate human benevolence and judge if the learner is in a state of knowledge or ignorance.

Keywords: **--e-Learning, Information Retrieval, Formal Language, Fuzzy Automata, Fuzzy Logic, Spell Check, Spelling Errors.**

Introduction

Advancements in computational power and technology have been continuously evolving and changing the way we interact with our surroundings. Such rapid improvements have allowed education, learning and academics to move beyond physical classrooms, geographical boundaries, or constraints of space and time. The learners do not need to move out of their comfort zone, instead they have experienced experts delivering lucid guides at a pace comfortable to the learner. All this available anytime, anywhere, thanks to e-Learning. While the use of consolidated models, pedagogically correct learning methodologies, and systematic observation has improved the quality of learning experience for the students, the full potential is yet to be achieved and these are still early days for e-Learning [1]. No matter how advanced and brilliant the content delivery might be, no learning system can be complete without effective and credible evaluation and assessment. The biggest challenge in an e-Learning environment is the assessment of the learners' achievement of learning objectives and their evaluation by teachers through an established process. The problem is further compounded because of the need to minimize human intervention in the evaluation process, to ensure the quality of evaluation is uniform for all the learners and can be completed within a short span of time. This necessitates the need to have an automated evaluation system which is smart and intelligent enough to replace the human evaluator. Such a system should then have human like qualities, like knowledge of subject, knowledge of expression, knowledge of learners' background, prudence to judge the learners' ability and benevolence to award them despite acceptable mistakes. Conventional learners' response analysis comprises of gathering learners' responses to questions put forward by the tutor, extracting the sense out of it and evaluating the same with respect to some predetermined benchmark. The picture however is different in case of an e-learning system because of the inability of the computer to emulate, completely, the human evaluator and the human intelligence. In this case it would require a learner to reproduce verbatim the contents delivered and thus encouraging rote learning [2]. Thus, the advent and extensive usage of MCQs to judge the achievement of learning objectives. The salient points contributing towards the popularity of MCQ's may be its objectivity, quantifiability and user friendliness and that they provide scope for more effective feedback to be targeted [3]. But, such a system not only is limited in its ability to test a learner's all-round knowledge, but also suffers from the serious drawback that it is possible to fake responses and fool the system and still have a fair chance of getting through with a good score. The fact that a person having 20% subject knowledge can effectively score 40% marks in a paper having MCQ's with four options, is disturbing [4]. Unlike most language processors, the approach of ruling out errors by hinting the correct spelling cannot be employed in learning scenarios [5]. In this paper, we propose a method that targets single word typed-in responses from the learner and evaluates whether the learner is in a state of knowledge or in a state of ignorance. The element of a human evaluator is achieved by employing a fuzzy automata inspired similarity based measure which is intelligent enough to disregard inadvertent errors committed by the learner. The benevolence of the system lies in its ability to partially award marks to a learners' response based on its degree of correctness without completely accepting or completely rejecting the provided response. In the process vying away from traditional binary logic and embracing fuzzy logic.The outcome of such a scheme is similar to that of a human evaluator who would consider responses with minor unintentional errors.

Related Work

Our work builds on significant reports from early days of e-learning, and the full potential in this field is yet to be achieved. Here we present some selected works relevant to our proposal. Sukkarieh et al. [6][7] reported NLP based techniques using POS taggers and various structured grammars, showing interesting results in automated marking systems. However, it lacks the humane evaluation factor and was computationally complex for practical usage. Natural language Dialog based interaction was proposed by Evens et al. [8] and further refined by Graesser et al. [9] use a coordinated dialog-based system where the learner and the pedagogical agent communicate over lesson content and world knowledge. LSA was used to gauge the proximity of the topics. Close ended test items can be presented in various ways, namely, MCQs, fill-in the missing words, matching the columns etc. proposed by Johnson et al. [10]. Chakraborty and Roy [11] have proposed a strategy using Artificial Neural Network to handle single word text responses. This model however uses extensive training methods which prove to be inefficient. Sahaet. al. [12] also proposed an effective scheme for single word textual responses based on Fuzzy Automata and scoring Heuristics. The method however could not handle insertion and deletion errors.

Smart Response Analysis

By now, it is obvious that text-based responses are a better way to test a learners' knowledge than MCQ or objective based questions. However, the inherent complexities of natural language processing have proved to be detrimental for the evaluation of text-based responses. Consequently, matching the expertise and effectiveness of an expert human tutor in evaluating text-based responses has proved to be computationally demanding and elusive. In the real-life scenario of a test in the e-Learning environment, the problem is augmented manifold since the learner may commit unintentional mistakes while responding to the system's queries. We want to look at such mistakes (typically spelling mistakes in our environment) benignly while being strict with the lack of knowledge. Learners may commit inadvertent spelling and grammar errors due to typographic mistakes, phonetic differences and such under the duress of testing situations. Such situations and feasible solutions have been discussed in our earlier works [12]. This paper builds upon those ideas and proposes a scheme which performs better with added refinements.

A. Problem Definition

We intend to design a system that accepts one-word answers from the learner in response to the system's queries. Next, the system should be able to evaluate the response and decide whether to accept or reject the response, considering that the learner may commit inadvertent mistakes in spelling. If the response is to be accepted, such that the errors are within an acceptable threshold, the system would need to provide a scoring / evaluation of the response. Formally, the problem may be defined as Let r = a_1 a_2… a_n be a string of alphanumeric characters that corresponds to the correct response for a query by the system. There is a set A = {s1, s2…} of strings such that $s_i \neq r$, for each i. Each s_i is a variation of r (the correct response) which would be treated as acceptable by a human evaluator even though it might have some spelling mistake(s) within a tolerable limit. The set B of unacceptable responses is given by B = ∑ - A - {r} where ∑ is the universe of all words. Here, both the sets A and B are unknown at the onset since we do not have a pre-decided list of acceptable and reject responses for each query. Then, given an arbitrary learner's response x, the issue is how to establish the mapping

$$f(x) = \begin{Bmatrix} accept, & if\ x = r, or\ x \in A \\ reject, & if\ x \in B \end{Bmatrix}$$

B. Problem Analysis

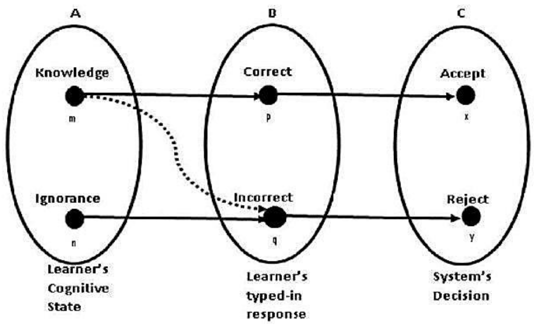

Fig. 1: Cognitive Model for Conventional Response Analysis

The cognitive model shown in Fig.1 captures the underlying essence of conventional response analysis in an e-learning system. The mappings m-p-x and n-q-y correspond to the usual scenarios of knowledge and ignorance. These mappings assume the response of the learner must be the true image of their state of knowledge, they either know or do not know the correct response, and the typed-in response reflects exactly the same. Consequently, such a response is either accepted or rejected by the system.

The challenge however, is introduced when the learner commits inadvertent mistake(s). Let us assume a situation where the learner knows the correct response but while typing makes some minor error(s) which a human evaluator would probably ignore. For example, if the correct response to a query is "compiler" and the learner types in "kompiler" or "conpiler" then such responses would probably be accepted by a human evaluator. However, a conventional automated response evaluator would reject such responses as incorrect and declare the learner to be in a state of ignorance (as shown by the dotted line m-q in Fig.1. To address this challenge, we propose a scheme for smart analysis of the learner's response. The purpose of this scheme is to introduce the latent mapping q-x as shown in Fig.1 such that the system is robust enough to withstand the petty errors that a learner may commit during their typed-in interaction with the system. The aim is to place the learner in his correct state of knowledge or of ignorance, benevolently accepting responses having inadvertent mistakes. The scheme is not strictly binary in nature (accept or reject responses), but uses concepts of fuzzy logic and aims to build a fuzzy automata inspired model that is capable of a partial marking scheme for partially correct responses. The learner is not given a complete free-hand, they are refused a perfect score due to benefit of doubt and marked according to their true performance. This impersonates the nature of a human evaluator who would accept responses with minor spelling errors, but at the same time penalize such a response by not awarding a perfect score. The present work considers evaluating textual responses with spelling erors to questions requiring only single word responses.

Basic Terminologies

A. Formal Language

A formal language is a set of words, i.e. finite strings of letters, symbols, or tokens. The set from which these letters are taken is called the alphabet over which the language is defined [13].

Formal Notions:

1. Alphabet: A set of symbols, indicated by V (e.g., V = {1, 2, 3, 4, 5, 6, 7, 8, 9}).
2. String: A string over an alphabet, V, is a sequence of symbols belonging to the alphabet (e.g., "518" is a string over the above V).
3. Linguistic Universe: Indicated by V*, denotes the set of all possible strings over V, included ϱ (empty string). The set V+ denotes the set V*-ϱ.
4. Grammar: It gives a Generative perspective: It defines the way in which all admissible strings can be generated.

B. Fuzzy Automata

A fuzzy finite-state automaton (FFA) is a 6-tuple, defined as

$M=<\Sigma, Q, Z, q0, \delta, \omega>$

where Σ is a finite input alphabet and Q is the set of states; Z is a finite output alphabet, q0 is an initial state, $\delta:\Sigma\times Q\times[0,1]\rightarrow Q$ is the fuzzy transition map and $\omega : Q\rightarrow Z$ is the output map.[14]

C. Fuzzy logic

Fuzzy Logic is a form of many-valued logic derived from fuzzy set theory to deal with reasoning that is robust and approximate rather than brittle and exact. Fuzzy [15] provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions.

D. Levenshtein Distance

In information theory, linguistics and computer science, the Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other. It is named after the Soviet mathematician Vladimir Levenshtein, who considered this distance in 1965. [16]Mathematically, the Levenshtein Distance between two strings a, b (of length |a| and |b| respectively is given by leva,b(|a|, |b|) where

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j)+1 \\ \mathrm{lev}_{a,b}(i,j-1)+1 \\ \mathrm{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

E. Types of Spelling Errors

The different types of spelling errors that can be handled by the system are listed in Table I.

F. Acceptable/Valid Errors in the System

In our effort to make the system sensitive and benevolent towards inadvertent mistakes, we introduce something called Valid / Acceptable Errors. A valid / acceptable error is when any of the errors listed in Table I comes from either of the two following sets, Valid Typographic Error Set (VTES) or Valid Phonetic Error Set (VPES).

The VTES takes its reference from adjacent keys of a standard QWERTY keyboard as shown in Fig. 2. So valid errors for the character 'S' would be any character from the adjacent set {A, D, W, X, Q, E, Z, X}

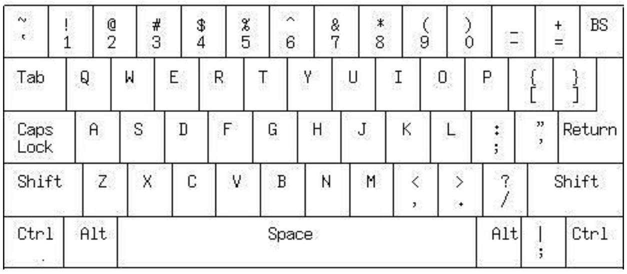

Fig. 2: Standard QWERTY Keyboard Layout

The VPES takes its reference from the Soundex Algorithm [17] which basically divides all the 26 characters from the english alphabet into 6 groups based on their phonetic similarity. The same is shown in Fig. 3.

Group 1: b f p v	Group 2: c g j k q s x z
Group 3: d t	Group 4: l
Group 5: m n	Group 6: r

Fig. 3: SOUNDEX Character Groups

Table I: Different Types of Spelling Errors

Sl. #	*Error Type*	*Explanation*
1	Apheresis	Omission of a letter from the beginning of the word. Eg: 'utomata' instead of 'Automata'
2	Apocope	Omission of a letter from the end of the word. Eg: 'Automat' instead of 'Automata',
3	Epenthesis	Insertion of extraneous letters to a word. Eg: 'Automasta' instead of 'Automata'
4	Homophone	Substitution of words which sound the same but are spelled differently. Eg: 'their' instead of 'there'
5	Keyboarding	Errors resulting from typing too quickly or carelessly and hitting nearby keys. Eg: 'Auyomara' instead of 'Automata'
6	Metathesis	Transposition of adjacent letters. Eg: 'Auotmata' instead of 'Automata'
7	Phonetic	Words which are spelled in a attempt to reflect their pronunciation. Eg: 'Kar' instead of 'Car'
8	Syncope	Omission of a letter from the middle of a word. Eg: 'Autmata' instead of 'Automata'

Solution Strategy

In the model scenario, a learner will be presented with a query from the system which has a one-word response. The learner will interact with the system by typing in their response via a standard keyboard. During their interaction with the system, learners are expected to commit unintentional spelling mistakes in their responses owing to duress, confusing pronunciation, or any other unforeseen factors. The system will take the learners response as input and proceed to evaluate the same. While evaluating the response, the system should be able to identify whether a mistake is unintentional or due to the lack of knowledge. In doing so, the system is expected to behave like a benevolent human evaluator who has the discretion to accept or reject a response from the learner based on the judgement of them being in a state of knowledge or in a state of ignorance.

We employ a two-fold solution strategy. In the first step, alearner's response is checked for the number and type of mistakes using a DamerauLevenshtein Edit Distance. A learner's response is rejected if there are more than X (max_edit, a value based on T and

N) number of edits in the response. Such responses are awarded 0 marks. Both T and N are discussed later. If a response is deemed to be acceptable, then the system must award some marks for the response. Marks lie between 0 and 1 (both inclusive) for each response. 1 indicates a completely correct response, 0 indicates a completely wrong response. The system may award any marks between the two extreme boundaries. However, for a response to be accepted it needs to achieve a minimum threshold mark T (which is variable and can be adjusted). If a response garners less than T marks, it is rejected, and the learner is awarded 0 marks. The awarding of marks is based on fuzzy membership inspired from Fuzzy Automata.

A detailed description of T, X, and other scoring factors are discussed in the next section.

System Design

Our system has a question bank and the correct response for each question is also known. An administrator can set the value for the threshold marks T. This determines the minimum marks a learner's response needs to fetch for it to be accepted. The value of T can be between 0 and 1(both inclusive). Lower values indicate a more benevolent tutor, higher values indicate a stricter tutor.

After this a learner is presented with a random question from the system and their response is recorded by the system. This response is then checked against the permissible maximum edits X. To identify the number of edits in the response from the original word, we use DamerauLevenshtein edit distance. This step also helps us to classify the type of errors (as shown in Table I) in the response. The value of X is determined using the formula

$X = floor\ (N/e^T)$

Where,

X: max_edit, the maximum number of permissible edits in the response

N: Length of the correct response in the system e: epsilon (2.71828)

T: Threshold as determined by the administrator

If the edit distance is more than X, then such a response is summarily rejected, and the learner is awarded 0 marks for such response.

If the number of edits is within permissible limits, then the response is evaluated and awarded marks based on the scheme discussed below.

• For every correctly matched character from the previous step, award 1/N marks and transition to the next state, such that if a response is completely correct, the learner will score 1 marks

For every substituted character (Keyboarding, Phonetic) from the previous step, we deal with it in two ways:

- o If the substituted character is from the VTES or VPES set, then we call it a valid substitution, add $(2T/(N\sqrt{N}))$ to the learner's score for each such occurrence, and transition to the next state. Note that this quantity is lesser than 1/N, which is awarded for correct characters
- o If the substituted character is not in VTES or VPES set, then we call it an invalid substitution, deduct $(T/(2*(N\sqrt{N})))$ from the learner's score for each such occurrence, and transition to the next state.

- For any transposition error (Metathesis) from the previous step, the pair is considered to be one edit. There is no invalid error in this case and such a pair is awarded $(2T/\ (N\sqrt{N}))$ marks while transitioning to the next state.

- For any insertion error (Epenthesis) from the previous step, no marks are awarded to the insertions. Each correctly matched character is still awarded 1/N marks and transition to the next state. However, the final marks is calculated out of N' (length of the response) and not N (length of the correct response)

- For deletion errors (Apheresis, Apocope, and Syncope) from the previous step, the scheme of handling them is slightly more complex. It is explained in the next section using an example.

For every response, the system checks each character of the response for correctness awards some marks (positive or negative) and moves to the next state. The process is repeated till no more characters remain in the response word. Fig. 3 shows different transition schemes under different scenarios.

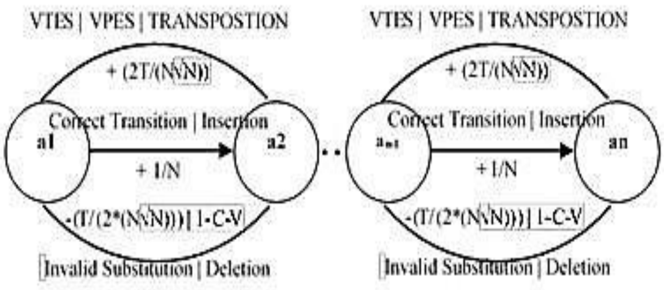

Fig. 3: Transition States of the System depicitng marking schemes for various situations

Results and Discussion

The system has been tested with close to 250 correct words and several different types of responses for each correct word. Below we show a few of them to highlight how different types of errors are handled and which responses are accepted, and which responses are rejected.

Case 1:

Let T = 0.7 Here N = 7 Then X = 3

Correct Word	*Response Word*	*Transition Marks*	*Explanation*
W	W	1/7	Correct Matching Input
E	E	1/7	Correct Matching Input
E	E	1/7	Correct Matching Input
K	K	1/7	Correct Matching Input
E	E	1/7	Correct Matching Input
N	N	1/7	Correct Matching Input
D	D	1/7	Correct Matching Input
Total Score		**1**	**Accepted**

The response is accepted with a full score of 1 since there are no errors.

Case 2:

Let T = 0.7 Here N = 7 Then X = 3

Correct Word	*Response Word*	*Transition Marks*	*Explanation*
W	W	1/7	Correct Matching Input
E	E	1/7	Correct Matching Input
E	*A*	0.07	VPES Substitution
K	K	1/7	Correct Matching Input
E	E	1/7	Correct Matching Input
N	N	1/7	Correct Matching Input
D	D	1/7	Correct Matching Input
Total Score		**0.92**	**Accepted**

The response is accepted (with a VPES Substitution Error) with a score of 0.92 since the score is greater than T=0.7

Case 3:
Let T = 0.7 Here N = 7 Then X = 3

Correct Word	*Response Word*	*Transition Marks*	*Explanation*
W	W	1/7	Correct Matching Input
E	E	1/7	Correct Matching Input
E	E	1/7	Correct Matching Input
K	K	1/7	Correct Matching Input
E	E	1/7	Correct Matching Input
N	N	1/7	Correct Matching Input
D	***B***	-0.01	Invalid Substitution
Total Score		**0.84**	**Accepted**

The response is accepted (with an Invalid Substitution Error) with a score of 0.84 since the score is greater than T=0.7. Note the final score of the response with invalid error is less than final score of response with acceptable / valid error.

Case 4:
Let T = 0.7 Here N = 7 Then X = 3

Correct Word	*Response Word*	*Transition Marks*	*Explanation*
W	W	N/A	Not Evaluated
E	***A***	N/A	Not Evaluated
E	***A***	N/A	Not Evaluated
K	***C***	N/A	Not Evaluated
E	E	N/A	Not Evaluated
N	N	N/A	Not Evaluated
D	***T***	N/A	Not Evaluated
Total Score		**0**	**Rejected**

The response is rejected with a score of 0, even though the errors highlighted are from VPES/VTES sets. The reason being that here number of edits = 4 which is greater than the permissible X. Thus, the response is summarily rejected without even evaluating fuzzy automata scores

Case 5:
Let T = 0.7 Here N = 7 Then X = 3

Correct Word	*Response Word*	*Transition Marks*	*Explanation*
W	W	1/7	Correct Matching Input
E	E	1/7	Correct Matching Input
E	E	1/7	Correct Matching Input
K	K	1/7	Correct Matching Input
E	E	1/7	Correct Matching Input
N	***D***		
D	***N***	0.05	Transposition Error Pair
Total Score		**0.77**	**Accepted**

The response is accepted (with a pair of Transposition Error) with a score of 0.77 since the score is greater than T=0.7.

Case 6:

Let T = 0.7 Here N = 9 Then X = 4

Correct Word	*Response Word*	*Transition Marks*	*Explanation*
A	A	1/7	Correct Matching Input
L	G		
G	L	0.06	Transposition Error Pair
O	O	1/7	Correct Matching Input
R	R	1/7	Correct Matching Input
I	I		
T	T	0.06	Transposition Error Pair
H	M		
M	H	0.06	Transposition Error Pair
Total Score		**0.61**	**Rejected**

The response is rejected (with 3 pairs of Transposition Errors) with a score of 0.61 since the score is lesser than T=0.7. Even though the number of edits was acceptable, the response is still rejected because it does not cross the minimum threshold marks.

Case 7:

Let T = 0.7 Here N' = 10 Then X = 4

Correct Word	*Response Word*	*Transition Marks*	*Explanation*
A	A	1/10	Correct Matching Input
L	L	1/10	Correct Matching Input
G	G	1/10	Correct Matching Input
O	O	1/10	Correct Matching Input
	O	0	Inserted Extra Character
R	R	1/10	Correct Matching Input
I	I	1/10	Correct Matching Input
T	T	1/10	Correct Matching Input
H	H	1/10	Correct Matching Input
M	M	1/10	Correct Matching Input
Total Score		**0.9**	**Accepted**

The response is accepted (with 1 Insertion Error) with a score of 0.9 since the score is greater than T=0.7.

Case 8:

This scheme handles deletion errors and is slightly more complex. All the positions in the word are marked in descending order, the word is initially scored a complete 1. Then marks are subtracted based on the position of the character which is missing if the character is a consonant, position value is ignored if it is a vowel. Let P represent position of each letter, N represent the length of the original word, and K represent the number of deleted vowel. Final score to be awarded is calculated as:

Score = 1- cons_neg – vowl_neg

Where

cons_neg = $(T*P) / (N\sqrt{N})$

vowl_neg = $(K2*T)/ (N\sqrt{N})$

Let T = 0.7 Here N= 9 Then X = 4

Correct Word	*Positional Index P*	*Response Word*	*Explanation*
D	9	D	Correct Matching Input
E	8	E	Correct Matching Input
T	7	T	Correct Matching Input
E	6	E	Correct Matching Input
C	5	C	Correct Matching Input
T	4	T	Correct Matching Input
I	3		Deleted Consonant
O	2		Deleted Vowel
N	1		Deleted Vowel
Total Score		**0.87**	**Accept**

cons_neg will be calculated for the character ‘N’ as (0.7*1) / (9√9)

vowl_neg will be calculated for the characters ‘I’ and ‘O’as (22*0.7) / (9√9)
score is calculated as 1 – cons_neg – vowl_neg = 0.87

So, this response will be accepted (with 3 deletion errors) and a score of 0.87 since it is greater than the threshold of T=0.7.

Conclusion

We have used DamerauLevenshtein Edit Distance to successfully identify the number and type of edits in the learners’ responses. If the number of edits was within acceptable limits we proceeded to use a fuzzy automata inspired technique to award marks to responses which were only partially correct. Based on the marks achieved by the response, we managed to identify whether such a response should be accepted or rejected, and with what score. Our system overcomes the drawbacks of a MCQ type testing scheme, by providing a more robust model scenario, a system which cannot be defeated by mere guess work. The system also introduces the elements of human intelligence, benevolence and knowledge while evaluating a learner’s response and not just their proficiency in the medium of answering. The system is able to overcome the mere mechanical techniques of evaluating a learner’s responseThis paper outlines only a part of our broader work in progress which will be able to handle phrase and paragraph responses from a learner by using vector space similarity indexing and latent semantic analysis.

References

[1] Koohang, A., Durante, A. (2003). Learners’ perceptions towards the web-based distance learning activities/assignments portion of an undergraduate hybrid instructional model. Journal of Information Technology Education, 2, 105–113.

[2] Chakraborty, U.K., Chowdhury, S., Roy, S., An Adaptive Scheme for Handling Deletion Spelling Errors’ for an Intelligent e-Learning System, Proceedings of 2nd International Conference on Computer Technology and Development, 2010, Cairo, Egypt, pp-260-263.

[3] Ramesh, S. Sidhu, S.M., Watugala, G. K. (2005). Exploring the potential of multiple choice questions in computer based assessment of student learning. Malaysian Online Journal of Instructional Technology, 2(1), 1–15.

[4] Bush, Martin Alternative Marking Schemes for On-Line Multiple Choice Tests, http://www.sbu.ac.uk/~bushm

[5] Kukich, K., Techniques for Automatically Correction Words in Text, ACM Computing Surveys, Vol. 24, No. 4, December 1992, pp. 377 – 439.

[6] Sukkarieh, J. Z., Pulman, S. G., Raikes, N. (2003) Auto-marking 2: An update on the UCLES-Oxford University research into using computational linguistics to score short, free text responses. http://www.cs.ox.ac.uk/files/237/AUTOMARKING2.ht m.

[7] Pulman, S. G., Sukkarieh, J. Z. (2005). Automatic short answer marking.Oxford University, Oxford, Workshop on Building Educational Applications Using NLP.

[8] Evens, M. W., Brandle, S., Chang, R.-C., Freedman, R., Glass, M. (2001). CI R C S I M-Tutor: An intelligent tutoring system using natural language dialogue. 12th Midwest AI and Cognitive Science Conference, Oxford OH, pp. 16–23.

[9] Graesser, A. C., Person, N., Harter, D., the Tutoring Research Group. (2001a). Teaching tactics and dialog in AutoTutor. International Journal of Artificial Intelligence in Education, 12, 257–279.

[10] Johnson, W., Russel, S., Nicholas, A., Clanton, E. S. (1974). Effects of alternative positioning of openended questions in multiple-choice questionnaires. Journal of Applied Psychology, 59(6), 776–778.

[11] Chakraborty, U.K., Chowdhury, S., Roy, S., Fuzzy Automata Inspired Intelligent Assessment of Learning Achievement, Indian International Conference on Artificial Intelligence 2011

[12] Saha, A., & Banerjee, P. (2015). Benevolent One-Word Text Response Testing System. The 2nd International Conference MICRO-2015 Conference Proceedings, 2, 113-118.

[13] Linz, P., An Introduction to Formal Languages and Automata, 4th Edition.

[14] Wen., Mo Zhi, Min., Wan, Fuzzy Automata Induction using Construction Mathod, Journal of Mathematics and Statistics, (2):2, 2006, pp.-395-400.

[15] Wang, Paul P., Da Ruan, Etienne E.Kerre, Fuzzy Logic: A Spectrum of Theoretical and Practical Issues.

[16] Levenshtein, V.I., Binary Codes Capable of Correcting Deletions, Insertions and Reversals, Soviet Physics Doklady, Vol. 10, p.707.

[17] Manning, C., Raghavan, P., Schutze, H., Dictionaries and Tolerant Retrieval in An Introduction to Information Retrieval, Cambridge UP.