AUTOMATED BENEVOLENT TEXT EVALUATION

Progress Report

In fulfilment of the requirements for the

NU 302 R&D Project

At NIIT University

Submitted by

Bandi Dinesh Kumar

Mrinal Pande

Pushkar Nagpal

Ronak Jain

Saurav Singh

B.Tech-CSE

NIIT University

Neemrana, Rajasthan

CERTIFICATE

This is to certify that the present research work entitled " Automated Benevolent Text Evaluation" being submitted to NIIT University, Neemrana, Rajasthan, in the fulfilment of the requirements for the course at NIIT University, Neemrana, embodies authentic and faithful record of original research carried out by Bandi Dinesh Kumar, Mrinal Pande, Pushkar Nagpal, Ronak Jain and Saurav Singh, student/s of B Tech (Computer Science and Engg.) at NIIT University, Neemrana. They have worked under our supervision and that the matter embodied in this project work has not been submitted, in part or full, as a project report for any course of NIIT University, Neemrana or any other university.

Prof. Aritra Saha
(Assistant Professor)
NIIT University

LIST OF FIGURES AND TABLES

LIST OF FIGURES

LIST OF TABLES

CONTENTS

INTRODUCTION

With the advancements in the Educational Sector indicate that the guided feedback and scaffold training is an effective method of instruction. To achieve these, then the correct analysis of the responses given by learner and proper evaluation becomes important. There are two kind of approaches which can be effective to build dialogue based systems i.e Machine Learning and NLP(Natural Language Processing). However, the efficiency and effectiveness of such systems depend on the accuracy of the underlying language processing unit. This kind of application is found easily in e- learning tests or assessments conducted for the learner's where the response is treated as the answer to the question asked and the evaluation is done on the basis of the entered response by the learner.

Conventional learners' responses analysis comprises of gathering the responses of the learner for the questions that have been put forward by the examiner, withdrawing the sense out of it and evaluating the same with respect to the default value set up by the human evaluator. But if we look towards the E-learning platform then it is a bit different from the conventional method because of the inability of the computer to follow, completely, the examiner and the human intelligence. As, a result, evaluating the response of the learner with respect to the predefined benchmark by the evaluator transpires. This would require a leader to reproduce verbatim the contents delivered and thus encourages rote learning.

Some popular tutoring systems employ similarity measure based approaches. A similarity measure based retrieval strategy is an algorithm that takes a query Q and a set of documents D1, D2, D3, ..., Di and identifies the Similarity Coefficient SC(Q,D) for each Di. Here if D1 represents the correct answer as provided by the system and D2 the answer provided by the learner, the proximity of the learners' response to the actual answer can be measured. The different identified models include but are not limited to the Vector Space Model, Probabilistic Model, Language Model, ANN (Artificial Neural Network) Model, Latent Semantic Indexing, Boolean Indexing etc. [1]. An important assumption taken however, in case of the design of the all such earlier techniques is the syntactic correctness of the learners' response. It would be over ideal to assume that the learners' responses will always be free of linguistic or spelling errors. Unlike most language processors, the approach of ruling out errors by hinting the correct sentence construction or correct spelling cannot be employed in learning scenarios.

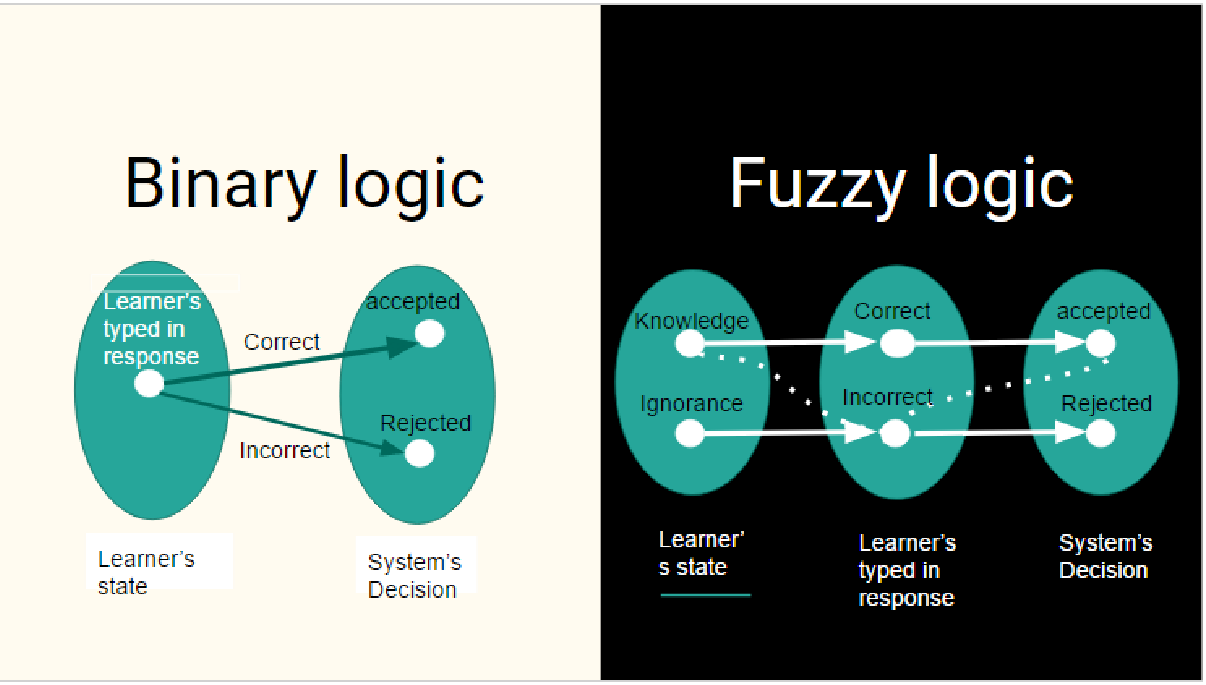


Fig1: Fuzzy logic

In this paper, we proceed to propose a method that introduces the element of a benevolent human evaluator employing a similarity based measure which is sensitive to the dissimilarities caused by the inadvertent errors committed by the learners on an E-learning platform. The remaining of this section is dedicated to introducing some terms used in the paper. This is followed by the problem definition, analysis and the solution strategy. The next section discusses the system design followed by a section each on results and discussions, future scope and conclusions.

ABSTRACT

Existing text evaluation systems can evaluate MCQ(multiple choice questions) using OMR(Optical Mark Reader) technology or using typed in answers in the form of choices from a list of answers.

But MCQs aren’t so apt anymore, they follow a binary system of giving an answer a 0 or a 1 score, hence it doesn’t really evaluate whether the student has knowledge or not.

The answer to making a system better than an MCQ system, follows out to be textual answers, but evaluating them is the next major problem, as unlike MCQs they require more than just a Correct or an incorrect logic and need to be properly evaluated and a system needs to be developed, hence the relevance of our project and this paper, to build such a system.

This paper presents an efficient methodology for the intelligent recognition of learners’ text-based responses over an e-learning environment. The method here targets the typed-in single-word and single sentence responses with the e - learning system, or during a computer-aided test. This proposed system is intelligently adaptive to the mistakes committed by the user while responding to the queries of the system. In this report, we proceed to propose a method that introduces the element of benevolent human evaluator system.

REVIEW OF LITERATURE

During the phase of the research and development we initially spent time to think about how to proceed with the work, then our mentor guided us and gave us the weekly tasks so as to get the knowledge of the topic and with the help of that we developed two approaches.

Then we referred to the research papers which were there on this topic for the initial phase so as to check that how we should make our project more updated than these presented in the papers.

The first research paper we went through, *Benevolent One-Word Text Response Testing System - Aritra Saha, Pritha Banerjee,* taked about algorithms based on single word evaluation schemes and used rescognized datasets to compare and show acceptability of different words.

The second paper, *A Novel Semantic Similarity Based Technique for Computer Assisted Automatic Evaluation of Textual Answers - Udit Kr. Chakraborty, Samir Roy and Sankhayan Choudhury* also talked about one-word evaluation and checked for pre and post expressions and dealed with stop words with conjunctions.

The next paper, *Intelligent fuzzy spelling evaluator for e-Learning Systems - Udit Kr. Chakraborty & Debanjan Konar & Samir Roy & Sankhayan Choudhury*, talke about evaluating words and implementing network map to a scheme to analyse and evaluate long textual answers written over learning platforms and evaluate them.

Hence we used this knowledge set and implemented more better algorithms to achieve optimum results and make a benevolent system for textual evaluation removing human evaluation barriers from the middle and improvising on systems like MCQ scoring.

OBJECTIVES

The objective of this project is to remove the MCQ system because from our perspective we think it doesn’t justify the examiner whether the learner has proper knowledge of that particular topic or not. So as to make advancements in the scoring scheme and to make the examinations more meaningful we have developed a system which can calculate a score of a word, phrase or a sentence, when entered by the learner during a descriptive examination and then the system will be able to calculate the score of that sentence w.r.t to the model answers provided by the examiner for the different marking schemes of that particular answer.

We currently, have fully developed this system which will focus on one sentence answer or short phrases or sentences. This system will also reduce the effort of human interference even in online based examination so this is why it is called as benevolent system.

METHODOLOGY

In our project we undertook various approaches for textual evaluation.
The initial approaches focused on single words and evaluating them to determine its correctness and reduce human effort and giving a suitable score to the word. Removing the binary factor of 0s and 1s in scoring a response by testing the user's intelligence and thus giving some partial marks for a partially correct answer.

Approach 1: Single word evaluation with step-up scoring

In our first approach, we made an algorithm to evaluate single words and the algorithm works in such a way that, it goes letter by letter and the word to be evaluated is compared with the correct word and 2 kinds of errors are undertaken.

1. Substitution: This is a typographical error in which the user may press the adjacent key in a hurry, so we address this issue and give a partial score to the user.
2. Transposition: This also is a typographical error in which the user may replace two adjacent letters of the word in a hurry, so such a response will again get a partial score.

We include arrays for representing the keyboard here in different lists to evaluate typographical errors in the word, the words are compared and a suitable score is obtained by comparing the letters and checking for errors, stepping up the score from a zero value.

Hence as we see this code will give us a score for the answered response by comparing it with the correct answer considering the mentioned errors and the scoring is a step-up scoring i.e. it starts from a zero and goes up with each letter evaluated.

Approach 2: Single word evaluation with step-down scoring

In this second approach we improved the previous system, we made an algorithm which evaluates the words considering the substitution and transposition errors but with a step down scoring i.e. It starts the score as a (say) 100 and steps it down, reduces it as it encounters an error or a completely incorrect letter thus providing the evaluator with an option to select a threshold value for textual evaluation. As soon as the score steps down below the threshold value the word is rejected.

So in the above algorithm, we added a threshold factor for evaluation and furthermore refined the scoring algorithm itself to achieve optimum scores for the received responses. We hence make a system that is a bit more efficient and follows up on the previous approach making it to stop when the score drops too low.

But both of the above approaches lacked in a big way as it didn’t consider any special cases with the responses, say if the correct word is “encrypt”, and the received response is “decrypt” , both of the above algorithms would have given the word a passable score even though it is a completely opposite answer.
Hence we researched and followed up to our next approach.

Approach 3: Textual evaluation using Naive-Bayes classifier

In this approach for textual evaluation, the algorithm used an encoding pattern for the received response, wherein the response was compared with the correct word and an encoding for the word, letter by letter was generated where the encoding was done by:

- 0 - Letters matched
- 1 - Substitution error
- 2 - Transposition error
- 3 - Incorrect letter

So following an encoding system, the system became more efficient. Now, such encodings were generated for different lettered words and different CSVs (Comma separated value files) were generated for different length words to make a dataset for later use.

Now once the csv files are ready for use, the main program can be executed. In the main program, there are sample question answers expecting a single word response from the user.

When the user enters a response, an encoding is generated for the response using the correct word.
Now, what the Naive Bayes classifier here does is, it opens up the corresponding lengthed csv file for the response encoding, and the encoding is modelled against the entire dataset from that CSV and a response 0/1 is predicted. This determines whether the received word is accepted or not. Furthermore a score is also generated for the response when its encoding is generated to know the kind of acceptance threshold for the response.

Approach 4: Textual evaluation using tf-idf scores for Phrases

Now that we made an algorithm for words, we moved on to develop a system for phrases and sentences.
In this system, we had model answer sets for a question and the user's response was compared against these model data sets and tf-idf scores are generated to evaluate which model answer is it closest to.

Also, to make the system more efficient, we have removed stopwords from the dataset, to leave them during tfidf calculation for better results. These stop words have been selected by a recognized English stopword set from *nltk_data.*

Some sample model answer:
"tf–idf, is a numerical way of finding how important a word is to a document. The term frequency is how many times the term is there in the document. The inverse document frequency means whether the term is common or rare across all documents it is calculated by dividing the total number of documents by the number of documents containing the term."

"tf-idf, is a numerical way of finding a word's importance in a document. The term frequency tf(t,d) is how many times the term is there in the document i.e. the number of times that term t occurs in document d. The inverse document frequency means how common or rare the term is in all documents."

User's response set:
"*The term frequency tf(t,d) is how many times the term is there in the document, the number of times that term t occurs in document d."*

RESULTS

Approach 1: Single word evaluation with step-up scoring

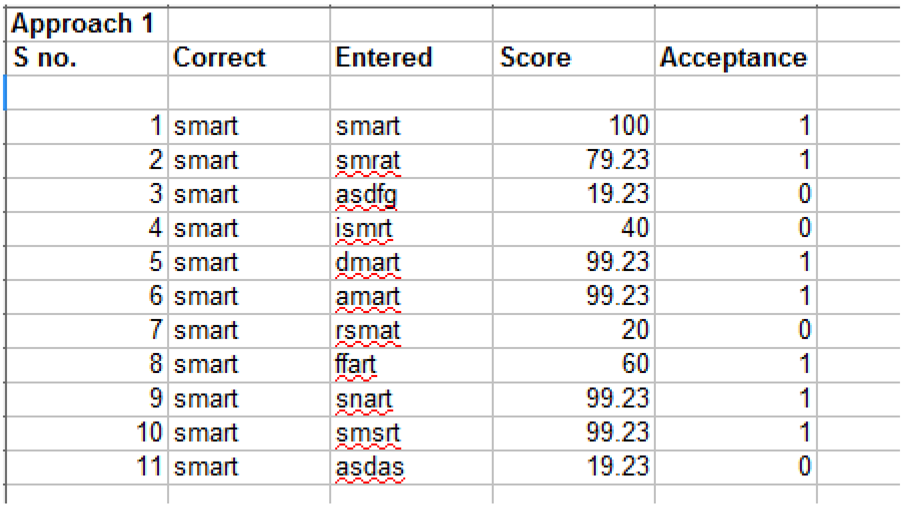
Objective:

The objective of these approach was to award marks on the basis of correctness.

Procedure:

The word was traversed letter by letter and correctness was awarded on substitution and transposition errors.

Here the person can set the acceptability of the score.



Approach 1				
S no.	Correct	Entered	Score	Acceptance
1	smart	smart	100	1
2	smart	smrat	79.23	1
3	smart	asdfg	19.23	0
4	smart	ismrt	40	0
5	smart	dmart	99.23	1
6	smart	amart	99.23	1
7	smart	rsmat	20	0
8	smart	ffart	60	1
9	smart	snart	99.23	1
10	smart	smsrt	99.23	1
11	smart	asdas	19.23	0

Fig2: Approach 1 output

Approach 2: Single word evaluation with step-down scoring

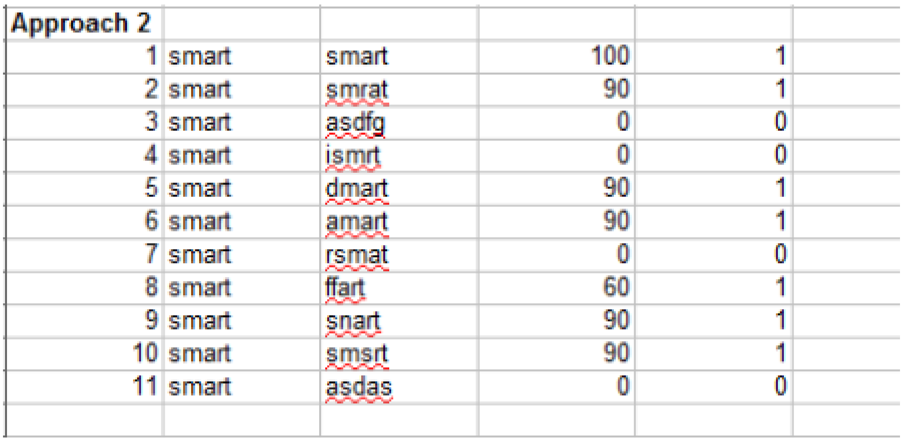
Objective:

The objective of the approach was similar to the first approach but instead we gave a word full score and penalized for wrong answer.

Procedure:

The word was traversed letter by letter and the score was deducted on substitution and transposition errors.

A threshold value of 50% is set and as the score goes below 50, the answer is rejected and a score of 0 is given.



Approach 2				
1	smart	smart	100	1
2	smart	smrat	90	1
3	smart	asdfg	0	0
4	smart	ismrt	0	0
5	smart	dmart	90	1
6	smart	amart	90	1
7	smart	rsmat	0	0
8	smart	ffart	60	1
9	smart	snart	90	1
10	smart	smsrt	90	1
11	smart	asdas	0	0

Fig 3: Approach 2 output

Approach 3: Textual evaluation using Naive-Bayes classifier

Objective:

Naive Bayes Works on numbers and probabilities.

Procedure:

We had to Encode the data using basic encoding for Naive bayes to work.

This encoding was on the basis of previous approaches i.e. correctness of the word.

Following the pattern of encoding that we have used :

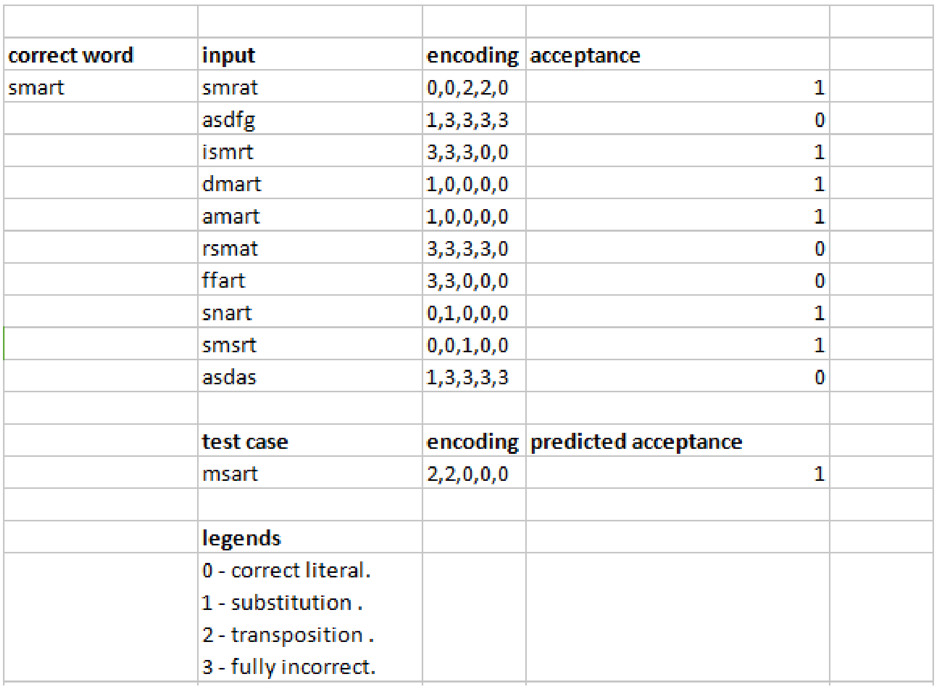
Encoding:

0- Correct

1- Substitution

2- Transposition

3- Incorrect



correct word	input	encoding	acceptance
smart	smrat	0,0,2,2,0	1
	asdfg	1,3,3,3,3	0
	ismrt	3,3,3,0,0	1
	dmart	1,0,0,0,0	1
	amart	1,0,0,0,0	1
	rsmat	3,3,3,3,0	0
	ffart	3,3,0,0,0	0
	snart	0,1,0,0,0	1
	smsrt	0,0,1,0,0	1
	asdas	1,3,3,3,3	0
	test case	**encoding**	**predicted acceptance**
	msart	2,2,0,0,0	1
	legends		
	0 - correct literal. 1 - substitution . 2 - transposition . 3 - fully incorrect.		

Fig 4: Approach 3 output explained

Procedure:

We have created a dataset which includes 10 correct and 10 incorrect sets of data for that particular correct answer.

Eg. For 3 letter word:

Correct 3 letter word CSV file:

Word : BAT

Acceptance	Encoding	Inputted word
1	000	bat
1	010	bst
1	100	vat
1	022	bta
1	001	bay

Table 1: 3word.csv sample for accepted input

Incorrect 3 letter word CSV file:

Word : BAT

Acceptance	encoding	Inputted word
0	100	nat
0	300	cat
0	003	bal
0	300	hat
0	330	lit

Table 2: 3word.csv sample for rejected input

For 4 letter word:

Correct 4 letter word CSV file:

Word : Fish

Acceptance	**Encoding**	**Word**
1	**0000**	**fish**
1	**0010**	**fiah**
1	**0300**	**fesh**
1	**0030**	**fich**

Table 3: 4word.csv sample for accepted input

Incorrect 4 letter word CSV file:

Word : fish

Acceptance	**Encoding**	**Word**
0	**0100**	**fush**
0	**1000**	**dish**
0	**0011**	**fiag**
0	**3333**	**phis**

Table 4: 4word.csv sample for rejected input

For 5 letter word:

Correct 5 letter word CSV file:

Word: Chair

Acceptance	Encoding	Word
1	00000	chair
1	22000	hcair
1	00220	chiar
1	00300	cheir

Table 5: 5word.csv sample for accepted input

Incorrect 5 letter word CSV file:

Word: Chair

Acceptance	Encoding	Word
0	03333	cnwkf
0	11333	vjzjd
0	33333	dnxlg
0	33333	oefms

Table 6: 5word.csv sample for rejected input

For 6 letter word:

Correct 6 letter word CSV file:

Word: Prompt

Acceptance	Encoding	Word
1	000000	prompt
1	002200	prmopt
1	030300	pmorpt
1	133300	opmrpt

Table 7: 6word.csv sample for accepted input

Incorrect 6 letter word CSV file:

Word: Prompt

Acceptance	Encoding	Word
0	331100	pfknot
0	133010	odlmot
0	333303	lfkjpg
0	333033	pglkph

Table 8: 6word.csv sample for rejected input

For 7 letter word:

Correct 7 letter word CSV file:

Word: Furious

Acceptance	Encoding	Word
1	0000000	furious
1	1000000	durious
1	0220000	fruious
1	0000300	furiaus

Table 9: 7word.csv sample for accepted input

Incorrect 7 letter word CSV file:

Acceptance	Encoding	Word
0	3000000	curious
0	1000000	gurious
0	0000330	furiyss
0	0003001	furkoua

Table 10: 7word.csv sample for rejected input

For 8 letter word:

Correct 8 letter word CSV file:

Word: Analysis

Acceptance	Encoding	Word
1	00000000	analysis
1	10003331	snaliysd
1	01000000	amalysis
1	01110010	abskysos

Table 11: 8word.csv sample for accepted input

Incorrect 8 letter word CSV file:

Word: Analysis

Acceptance	Encoding	Word
0	03030001	ajaoysid
0	33333333	osdfmodl
0	33333033	wekaksfn
0	00333331	anmzalld

Table 12: 8word.csv sample for rejected input

Approach 4: Textual evaluation using tf-idf scores for Phrases

Objective:

Using the tf-idf evaluation technique we will calculate the score of the short phrase given by the learner.

Procedure:

Tf scores are calculated by:

$$tf_{t,d} = 1 + \log tf_{t,d}$$

Where **(t,d)** = terms in the learner's answer in each document

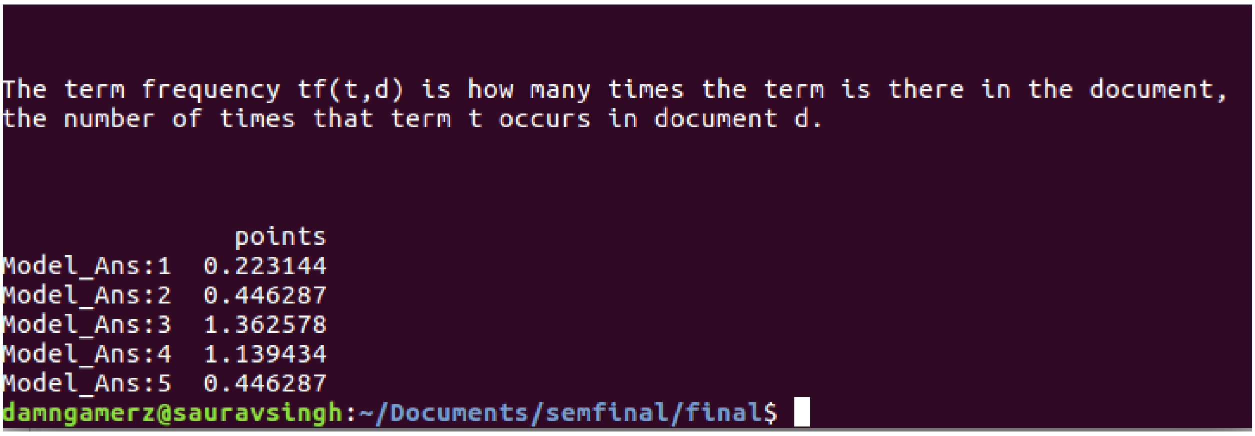
Idf scores are calculated by:

$$IDF(t) = \log(N / Df)$$

Where N = Total number of documents.

df= Number of documents with term t in it.

Using these scores the response is evaluated and a suitable model answer is recieved as the best match thus awarding it a score for the same.



```
The term frequency tf(t,d) is how many times the term is there in the document,
the number of times that term t occurs in document d.



               points
Model_Ans:1  0.223144
Model_Ans:2  0.446287
Model_Ans:3  1.362578
Model_Ans:4  1.139434
Model_Ans:5  0.446287
damngamerz@sauravsingh:~/Documents/semfinal/final$ 
```

Fig 5: Approach 4 output

CONCLUSION AND FUTURE WORK

We successfully made a system for textual evaluation for words, phrases and short sentences through our approaches in the project. Making it a benevolent system at the same time to test the user's knowledge set, implementing a fuzzy logic system to make it better.

In the future we intend to implement approaches to make our algorithm work for proper paragraphs and entire documents, making it benevolent at the same time reducing human effort and making an optimum algorithm for the same.