IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 3)
Available online at: www.ijariit.com

Answer script evaluator

Sachin B. S.
sachinbet17@gmail.com
Vidyavardhaka College of Engineering, Mysore, Karnataka

Somesh T.
somyash97@gmail.com
Vidyavardhaka College of Engineering, Mysore, Karnataka

Shivaprasad K.
shiva15997@gmail.com
Vidyavardhaka College of Engineering, Mysore, Karnataka

Sumanth Hegde
sumanthhegde1029@gmail.com
Vidyavardhaka College of Engineering, Mysore, Karnataka

Radhika A. D.
radhika.ad@vvce.ac.in
Vidyavardhaka College of Engineering, Mysore, Karnataka

ABSTRACT

Every college, university, school conduct exams and most important part of exams are the results. In order to get these results, the exam papers have to be evaluated one by one manually. This process of evaluating the exam papers is time consuming and requires more manpower. To overcome this solution, we have come up with a thought that removes the manual evaluation process. Our project focuses on developing a system that evaluates an answering script against a pre-uploaded marking scheme. Initially, a model is trained with a huge corpus (text data) and this model is used for further comparison of answer sheets with marking scheme. For evaluation, the marking scheme is processed and created a bag of words. In the same way, even answer sheets are processed and created a bag of words. Now, these 2 values are compared using the LSI model and marks are assigned to the answers according to the comparison percentage obtained.

Keywords— *LSI, TFIDF, Document similarity, Sentiment Analysis*

1. INTRODUCTION

In this era of technology and advancements, everything is being digitized. From online payments to patient's medical health records, everything is being digitized and made available online. However, there are still many problems unattended which don't have a digitized solution and continuous work is in progress to provide solutions to all these problems.

There are OMR sheets in which objective answers are selected from a given range of options. Evaluation of these sheets are also simple as they just check the orientation of the coloured dot and marks are awarded if the coloured circles are there for the correct options. These systems already exist and are already implemented long back.

However, there are no evaluations systems that are capable of evaluating subjective answers from a given range of answers in an answering script. We have designed and implemented a software program where the main job is to evaluate subjective answer scripts against a pre-uploaded marking scheme. Marking scheme contains the correct answers for each question of the question paper for which the student writes his/ her own answers.

2. EXISTING SYSTEM

The current system of evaluation of answer scripts involves manually checking all the answer scripts one by one and evaluating them by looking at the marking scheme. This system of evaluation takes a lot of time and also makes way for uncertain evaluation. It also takes a lot of time and manpower and involves each person to personally evaluate a specific subject answer paper one by one. The constraints in this system are time, manpower and efficiency. Efficiency depends on the number of people evaluating, number of papers, and the type of evaluators (different number of evaluators for different subjects). If a number of papers are huge and the number of evaluators is few, the efficiency is low. In the case of huge universities, this efficiency is low most of the times as there will be a huge number of students in the university and the number of answer scripts is also huge. In some universities, there is a system that uses scanned copies of the answer sheet instead of the physical paper itself. This provides an option to perform valuation digitally but it has to be done by a human.

3. PROPOSED SYSTEM

This system is an extension of digitization of answer sheets. In this system, along with digitization of answer sheets, even valuation is done by the system itself. The system performs extraction of keywords and this extracted text is then compared to the reference text using Latent similarity indexing. The inputs to this system are digital answer sheets and the reference text.

The main advantage of this system is the reduction of time of valuation drastically and it also diminishes man power. Using this system, many numbers of answer sheets can be evaluated (regardless of what subject they belong to).

This system can also be hosted online and the models to be used can be made available online so that the evaluator can be used in a wide range of devices and can be accessed remotely. Currently, in this project, the input answer script is limited only for textual data and does not handle images or equations.

4. METHODOLOGY

For this project to work, a model is trained with a huge set of data (about 60GB of a text file) that is., wiki_corpus so that our model has knowledge of a huge number of English words and it has a mapping from word to the number. This huge data is trained on LSI and an LSI model is created. After that, a bag of words is created for the marking scheme, this bag of words is the corpus. Now, this corpus is prepared for comparison by passing the corpus through the LSI model which generates a set of indexing values that eases the comparison. In the same way, a bag of words is created from the student's answer script and even this is passed through the LSI model and generated a set of values is passed through that indexing values created from the marking scheme and yields a comparison value.

The processing of input data into a bag of words and extracting and comparing the answer sheet and marking scheme happens in many stages. They are as follows:

4.1 Input text processing

This is the first and most important stage in our project as it determines the text that is available for processing. We are considering digital format for both the answer sheet and marking scheme.

Both the marking scheme and answer sheet have to follow the below rules:

(a) Each answer must start with question number followed by ")" and then the text must follow.
(b) Each point/sentence must be ended with a full stop.
(c) At the end of each answer, the tilde (~) symbol must be entered after the last full stop.
(d) There must be 1 empty line between each answer.

Format of the answer sheet and marking scheme:

```
1)-----------------------------------------.
--------------Answer 1--------------------- .~

    >>>>one-line space<<<<

2)---------------------------------------.
--------------Answer 2----------------------.
---------------------------------. ~
```

4.2 Removing Stop Words

From both the marking scheme and the answer sheet, each answer is split to different lists. Answers are separated by a custom delimiter (~) tilde.

Now from each answer lists, each sentence is again put into another list. So the number of lists inside the answer will be the same as the number of sentences in the answer.

Stopwords are words such as "this, that, is, as" etc.
We remove stopwords from teaching sentence because these words do not contain any meanings on their own hence they are removed in this stage.

After removing stopwords, all that remains is keywords and meaningful data which can be used for comparing.

4.3 Bag of words

The problem with machine learning algorithms is that they are unable to process text data directly. The text must be converted to numbers for the algorithm to handle. Bag of words is a method of extracting features from the text.

It usually stores words in the form of a dictionary where the vocabulary of trained data is the key and its frequency will be their corresponding values.

TFIDF is a statistical measure used to evaluate how important a word is to document. The importance increases proportionally to the number of times a word appears in the document.

4.4 Comparison of text

This stage is the actual stage where the data comparison between the answer script and the marking scheme happens. We provide two extra files as input:

(a) Marks distribution file which contains question-wise marks distribution.
(b) Written answers list where it contains a list of questions that are attended by the student.

While performing the comparison, there arise 2 scenarios.

First, when the student has written less content than that of marking scheme. In this case, for each sentence is the answer sheet, all the sentences in the marking scheme are compared and the sentence (in marking scheme) which has maximum similarity is assigned to that particular sentence in the answer sheet. Similarly, each sentence is compared and gets its own comparison percentage.

Second, when the number of sentences in the student's answer sheet has more lines than that of the marking scheme. In this case, for each sentence in the marking scheme, all the sentences in the answer sheet are checked and sentence with a maximum comparison percentage is assigned to that particular sentence in the answer sheet. Similarly, all the sentences in the marking scheme are compared to the entire answer sheet and each line in the answer sheet will now be assigned to some comparison percentage.

4.5 Sentiment analysis

This is the stage where the polarity of the sentence is measured. The polarity of a sentence means the nature of sentence that is., negative or positive or neutral. This stage is there to provide negation detection for both marking scheme and answer sheets. For example, in the marking scheme, there is a sentence: "India is a secular country "but the student has written, "India is not a secular country". Then we must be able to differentiate these sentences and marks must not be assigned to those sentences which oppose the actual answer.

For the sentiment analysis, we use pattern library. It helps in extracting the polarity and subjectivity of a sentence. It also has a provision to check the polarity assessment that is., it shows the word which is responsible for the polarity in the sentence.

This stage is applied to the answer sentences only when the comparison percentage of that particular sentence is above 80%. Again we have 3 scenarios. The processed sentences cannot be used for sentiment analysis because stopwords have already been removed. So for this stage, we access the original sentences through its index. The sentence of answer sheet is taken and its most similar marking scheme sentence is taken. The polarity of both the sentence is checked.

In the first case, polarity is checked for both the sentences (one from marking scheme and other from answer sheet) and if both the polarities are either positive or either negative, then that sentence is retained in the answer sheet and the comparison percentage is finalized for that sentence.

In the second case, if the sentences have opposite polarity (one negative and one positive) then we take the difference of polarities. If the difference is less than 0.4, we retain the sentence in the answer sheet. If the polarity differences are more than 0.4, then that sentence is deleted from the answer sheet.

4.6 Assigning Marks

After removing all the sentences which have opposing polarities whose difference is greater than 0.4, the next step is to calculate the marks for the answer.

Each sentence has its own comparison percentage. To assign the marks for one particular answer, the sum of these comparison percentages is taken and is divided by the number of sentences in the marking scheme.

Now each answer will have its average similarity percentage and the marks is assigned based on this percentage. To get the total marks, marks assigned to all the answers are added and displayed against the maximum marks.

5. DESIGN AND IMPLEMENTATION

The entire stages and processing programs are written in python language and it also acts as the back-end for our project. Even front end is developed to ease the operation of this project and to provide a graphical interface to the user so it is easy to understand and simple.

Providing a GUI to the project also helps people with limited knowledge about running programs on the terminal and extracting the output by providing a more familiar click and run the interface.

The front-end for the project is a typical window which is designed and coded in wxpython, a python GUI library.

It also has a file dialogue so that it is easy to browse the files from the system and select them for evaluation. The entire background script is linked to a button so when the user selects all the files required and clicks that particular button, the program runs in the background, performs all the 6 stages and outputs the marks obtained on the GUI. The detailed question-wise marks distribution can be seen from the program's terminal window.

6. RESULT ANALYSIS

Though the machine learning concept is been in the market for over a decade, it still hasn't been widely used. The reasons behind this are:

(a) Implementing machine learning is not an easy task and it takes a lot of time for one to gain knowledge on the libraries that are used for machine learning.

(b) The accuracy of the future prediction of values depends highly on the data that is used to train the model. So finding the right dataset to train the model is very difficult and challenging.

For a sample social science answer script along with its marking scheme, we were able to get desired results. For testing purposes itself, some answers were written correctly and some answers were written wrong deliberately to analyze the results of the system. But we were able to achieve the desired results in all cases and it was found that the system was performing the evaluation pretty strictly. However, we can have a tolerance factor where in the comparison percentages can be rounded off to the next whole number or the next number which is divisible by 10. (For eg, if comparison percentage for a sentence is 73%, we around it off to 80 %.)

7. FUTURE ENHANCEMENTS.

This system currently is limited to only textual data. However, in the near future, the model can be made to handle and evaluate equations and diagrams.

For this system to work, we need the answers and the marking scheme in the digital format. An image processing model can be implemented so that the system can also accept handwritten data and convert that to digital data automatically.

This system can be deployed in a server local to college or a university so that all professors have access to the system and can evaluate with ease either with their computers or phones.

8. CONCLUSION

In the existing trend, the answer scripts are evaluated manually by checking line by line which requires more manpower and time.

But the model that we develop enables answer script to be evaluated at a very short time and doesn't require any human intervention. However, the model that is to be developed depends highly on the dataset that we use to train. Therefore, it is very important that we choose the dataset and the hyperparameters in such a way that maximum accuracy is obtained.

By using a good dataset, we can be able to increase the gap between the synonyms and the antonyms, thereby increasing the overall accuracy of the model.

9. REFERENCES

[1] "NLTK: The Natural Language Toolkit" authored by Steven Bird and Edward: Introduction to text processing

[2] "A Compare-aggregate Model For Matching Text Sequences" authored by Shuohang Wang and Jing Jiang: matching vector values

[3] "Progress in the Application of Natural Language Processing to Information Retrieval Tasks" authored by Alan F. Smeaton: finding out the same meaning in a different set of words

[4] "An empirical evaluation of doc2vec with practical insights into document embedding generation" by Jey Han Lau and Timothy Baldwin

[5] Bag of words URL: https://en.wikipedia.org/wiki/Bag-of-words_model

[6] Gensim URL: https://radimrehurek.com/gensim/

[7] LSI Model and its implementation URL: https://radimrehurek.com/gensim/models/lsimodel.html

[8] The pattern for sentimental analysis. URL: https://www.clips.uantwerpen.be/pages/pattern-en

[9] TFIDF URL:https://medium.freecodecamp.org/how-to-process-textual-data-using-tf-idf-in-python-cd2bbc0a94a3

[10] GUI using wxpython URL: https://wxpython.org/

[11] Designing buttons and binding them with python functions. URL: https://www.tutorialspoint.com/wxpython/