**SUBJECTIVE EVALUATION OF ANSWERS**

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **REG.NO** | **E-MAIL** | **MOBILE NO** |
| Dhivyashri R | 2019103015 | Dhivyashriramesh@gmail.com | 7338774745 |
|  |  |  |  |
|  |  |  |  |

INSTRUCTOR: **Dr AROCKIA XAVIER ANNIE R**

Contents

[1.ABSTRACT 5](#_Toc105362927)

[2.INTRODUCTION 5](#_Toc105362928)

[3.LIT SURVEY 7](#_Toc105362929)

[4.PROBLEM STATEMENT 9](#_Toc105362930)

[5. PROBLEM SOLUTION 9](#_Toc105362931)

[5.1 Data Set:- 9](#_Toc105362932)

[5.2 Input :- 9](#_Toc105362933)

[5.3 Approach:- 9](#_Toc105362934)

[6. NOVELTY 11](#_Toc105362935)

[7. ARCHITECTURE 12](#_Toc105362936)

[7.1 ABSTRACT ARCHITECTURE 12](#_Toc105362937)

[7.2 EXPLANATION OF THE ARCHITECTURE 12](#_Toc105362939)

[7.3 TASK ARCHITECTURE 13](#_Toc105362940)

[7.4 EXPLANATION OF THE ARCHITECTURE DIAGRAM 14](#_Toc105362941)

[8. DETAILED MODULE DESIGN 14](#_Toc105362942)

[8.1 Image recognition MODULE 14](#_Toc105362943)

[8.2 PREPROCESSING Module 14](#_Toc105362944)

[8.3 PATTERN MINING MODULE 14](#_Toc105362945)

[8.4 SIMILARITY COMPARISON MODULE 14](#_Toc105362946)

[8.5 EVALUATION MODULE 14](#_Toc105362947)

[9. IMPLEMENTATION 14](#_Toc105362948)

[9.1 Initial Set-up: 14](#_Toc105362949)

[9.3 Tools: 14](#_Toc105362950)

[9.4 Code Snippets: 14](#_Toc105362951)

[9.4.1 14](#_Toc105362952)

[10. SUPPORT INFORMATION 14](#_Toc105362953)

[11. SURVERYED CONTENT 14](#_Toc105362954)

[12. RESULTS AND COMPARISON 14](#_Toc105362955)

[13. CONCLUSION AND FUTURE ENHANCEMENTS 16](#_Toc105362956)

[14. List of References 17](#_Toc105362957)

[15.Appendix A: 18](#_Toc105362958)

[16. Appendix B: References 19](#_Toc105362959)

[17.Appendix C: Key Terms 20](#_Toc105362960)

1.ABSTRACT

Evaluation of answer sheets is a tedious process. Though multiple-choice questions and objective questions have been shown to be easily evaluated automatically, the evaluation of subjective answers is still a problem that requires attention. Subjective evaluation of answers is a time and man-power intensive job. Therefore, the automation of this process will provide a solution to the problem of answer evaluation and will most definitely improve the ease of correction serving as either an aid or replacement for manual work. This development will try to remove the pre-existing constraint for automatic evaluation which only allow the evaluation of objective answers which can be fed to a machine to verify. This project aims to provide a method to evaluate a subjective answer by providing it a particular score.

This project aims to use pattern mining and similarity checking with the help of keywords and principles of Natural Language Processing to present a way to evaluate an answer subjectively. Natural Language Processing can be explained as the computational techniques used for the analysis and synthesis of natural language and speech. In simpler terms, it is the ability of a computer program to understand human language as it is written and spoken. By the means of this project, an objective has been set which is to use the similarity between the answer key provided by the teachers to the answers to establish a score for the respective answer. Along with this, some patterns common amongst the student answers are found alongside the count of these patterns. The proposed method would consist of pre-processing, pattern mining, similarity comparison, a weighted function generation and the actual evaluation of the answer. Once implemented this would aim the process of evaluation of a subjective answer by reducing the time and effort required to evaluate an answer by a human being.

2.INTRODUCTION

With the increase in the usage of technology in all domains of life, and in specific, the increase of automation in different fields, the education sector is also witnessing a great surge in the advent of technology to solve problems. The task of evaluating and correcting examination answers is particularly a task that is intensive on the system demanding time and labour. The current success of objective answer evaluation however limits the scope to which automation can be wielded. It does not permit the conduct of exams with subjective answers. To address these the project hopes to use text similarity and pattern mining along with a weighted function generation.

Text Similarity is one of the essential techniques of Natural Language Processing that is used to find the closeness between two chunks of text by its meaning or by surface. This has to be done by first pre processing the data or the text we have of the answers and later using a specific metric to analyse similarity. A machine understands only mathematics; therefore, we need a mathematical mapping to process the similarity. This project intends to use the cosine similarity metric to analyse similarity between the decided key of the teacher and the answer provided by the student. Cosine similarity is one of the metrics to measure the text-similarity between two documents irrespective of their size in Natural language Processing. A word is represented into a vector form. The text documents are represented in n-dimensional vector space. Once processed the answers are given a score according to both the commonalities and the similarities observed via the above the methodologies. With reference to this project the Natural Language Tool Kit of the Python programming language. The Natural Language Toolkit or NLTK is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language.

With the above description as an overview, the below text involves the module split up of the project and the walk through of the general procedure followed in the project

**There are 5 modules in our project**

1. Image Recognition
2. Pre-processing
3. Pattern Mining
4. Similarity Comparison
5. Evaluation

The proposed method would consist of pre-processing, pattern mining, similarity comparison, a weighted function generation and the actual evaluation of the answer. Once implemented this would aim the process of evaluation of a subjective answer by reducing the time and effort required to evaluate an answer by a human being.

**Natural Language Processing and Natural Language Toolkit**

On a daily basis, particularly in businesses and other domains, massive quantities of unstructured, text-heavy data are produced and it is therefore a significant need to find a way to efficiently process it. A lot of the information created online and stored in databases is natural human language, and until recently, businesses could not effectively analyse this data. This is where natural language processing is useful. Natural Language Processing is how computers are made to understand human language and speech. In general, for analysing paragraphs, one would require tasks such as segmentation, sentence breaking, lemmatization to observe key words, stemming to identify root words and much more. With reference to this project particularly we would need to make use of the task of lemmatization, tokenization, stemming in pre processing itself. These processes will be explained in a detailed manner in the following sections that talk about the process architecture of the project.

This project as mentioned before uses the Natural Language Toolkit for the accomplishment of its objective.  Natural Language Toolkit provides easy-to-use methods for tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries. This is a major advantage while taking up the task of text analysis and similarity analysis which would require stemming and separation of key words to analyse the score of a given answer. Owing to the fact that a score is determined by the presence of certain keywords analysed from text similarity, the natural language toolkit is the perfect tool to use for the selected task.

3.LITERATURE SURVEY

[A] Concept Mapping Method Text data analysis: Computer aided automated assessment system, Nisarg Dave; Harsh Mistry; Ai Prakash Verma Computational Intelligence Communication Technology (CICT), 2017 3rd International Conference Computer aided evaluation systems are generally considered objective types of questionnaires.

This paper is emphasizing the issues of computer aided automated assessment and proposing a model for handling these issues. Using such mechanism, a faculty can avoid the evaluation process manually. Students can be automatically graded using the application and given a summary report. After conducting many experiments, the results showed only 80 percentage of expectancy.

[B] Information Extraction Method A review of an information extraction technique approach for automatic short answer grading” Uswatun Hasanah ; Adhistya Erna Permanasari ; Sri Suning Kusumawardani ; Feddy Setio Pribadi Information Technology, Information Systems and Electrical Engineering (ICITISEE), International Conference. The requirement for automatic short answer grading (ASAG) system brings researchers to discover more knowledge about this field. Many techniques have been developed to reach the highest accuracy. It can be processed by following stages: creating data set, pre-processing, model building, grading, and model evaluation. One of the techniques which commonly used is information extraction technique. Information extraction is a technique that employing finding fact on the student answers as patterns and then matches these to the teacher answer. The accuracy is pointed out in computer and human agreement. The goal of this paper is to present a review of several ASAG research which using information extraction technique. However, this paper does not conclude the best method which can be used for general cases.

[C] Knowledge Analysis Method Automated question answering system using ontology and semantic role”S. Jayalakshmi ; Ananthi Sheshasaayee, Innovative Mechanisms for Industry Applications (ICIMIA), 2017 International Conference Semantic similarity is an essential part for question answering, it is used various fields such as Artificial Intelligence, Natural Language Processing, information retrieval, Document Retrieval and Automatic evaluations. This paper mainly focuses on similarity measure based on the posted query, and finding the appropriate meaning between the words. Accessing an accurate answer from the web document is challenging task. The proposed approach is used to analyse and measuring the similarity between the words. It presents the Web and Semantic Knowledge-Driven automatic question answering system (WAD). It encompasses three phases to enhance the performance of QA system using the web as well as the semantic knowledge. Initially, the WAD approach determines the user query, query expansion technique and entity linking method. The ontology-based information is used in WAD to rank the answers and experimental results provide the result with high accuracy than the baseline method. The drawback of this paper is that we need to require rich knowledge base for the evaluation purpose.

[D] Document Similarity Based Technique “Automatic Short –Answer Grading System (ASAGS)” P.Selvi, Dr.A.K.Bnerjee. Automatic assessment needs short answer-based evaluation and automated assessment. Various techniques used are Ontology, Semantic similarity matching and Statistical methods. An automatic short answer assessment system is attempted in this paper. Through experiments performed on a data set, we show that the semantic ASAGS outperforms methods based on simple lexical matching; resulting is up to 59 percent with respect to the traditional vector-based similarity metric.

OUR APPROACH

Our Proposed system uses an approach that combines the pattern mining unsupervised technique with the similarity measurement for the evaluation of descriptive answers. Using pattern mining, we can analyse enormous amounts of data quickly. The proposed system involves parsing of text and find the semantic meaning of student answer and finally assign the final scores using Natural Language Processing. Semantic Analysis helps machines interpret the meaning of texts and extract useful information, thus providing invaluable data while reducing manual efforts. In this project the similarity between the model answer and students’ answers, and the promotion of pattern matching among the students’ answers are processed with the NLP technique. Later on, the final scores are assigned. The proposed model has greater accuracy generated than other methods.

4.PROBLEM STATEMENT

**Subjective evaluation of answers using Natural Language Processing methods**

The manual system for evaluation of Subjective Answers for technical subjects involves a lot of time and  
effort of the evaluator. Subjective answers have various parameters upon which they can be evaluated such as the question specific content and writing style. Evaluating subjective answers is a critical task to perform. When human being evaluates anything, the quality of evaluation may vary along with the emotions of the person. This system can be used instead in order to reduce their burden. It will save a lot of effort and time on teacher’s part. The human efforts applied in this repetitive task can be saved and spent more in other academic endeavours. The obvious human mistakes can be reduced to obtain an unbiased result. The system calculates the score and provides results fairly quickly.  
  
There are a number of commercial assessment tools available on the market today; however, these tools support objective type question such as multiple-choice Questions or short one-line free text responses. This will assess student’s depth of knowledge only at lower level of taxonomy of educational objectives. They fail to assess student’s performance at higher level of taxonomy of educational objective. So, to overcome the problems encountered the proposed system involves parsing of text and find the semantic meaning of student answer and finally assign the final scores using Natural Language Processing

5. PROBLEM SOLUTION

By the means of this project, an objective has been set which is to use the similarity between the answer key provided by the teachers to the answers to establish a score for the respective answer. Along with this, some patterns common amongst the student answers are found alongside the count of these patterns. The proposed method would consist of pre-processing, pattern mining, similarity comparison, a weighted function generation and the actual evaluation of the answer. Once implemented this would aim the process of evaluation of a subjective answer by reducing the time and effort required to evaluate an answer by a human being.

## 5.1 Input

## Scanned handwritten images, model answer

## 5.2 Approach

The answer sheet image is uploaded as input. Using Python-tesseract, an optical character recognition (OCR) tool for Python and an open-source text recognition engine, text is extracted from images. Pytesseract , a wrapper for the Tesseract-OCR Engine is installed. We convert from image to string using the method image\_to\_string().we first converted the image to gray scale and then specified the kernel shape and size. Next, we found the contours and looped over them, chopping the rectangle area. Next, we have passed the rectangle area onto pytesseract for extracting text from it and then writing it into the text file.

Text file is pre-processed, where Sentence Segmentation, Tokenization, Removing Stop Word, and Word Stemming is done. Tokens are generated after pre-processing. The sequential pattern is generated from the tokens of each sentence. The count for each obtained pattern is also determined. The cosine-based similarity approach is used to calculate the similarity between the input answer and the model description. To compute cosine similarity between two sentences sentence number one and sentence number two, the sentences are turned into terms or words, words are transformed in vectors. The given answers are allotted a score by identifying both commonalities and similarity between the students’ answers. Finally, scores are printed.

6. NOVELTY

Big Data Analytics on any structured data is always a task comprising higher implementation challenges and novel ideas when compared to unstructured data. **Our deployment on a cluster provides a user with our framework to compute maximum-flow on any graph including parallel edges and self-loops**. Our novelty constitutes of relating this to real-world problem where we believe the maximum traffic-flow value can be added as a quality parameter in large networks. **Also the same output can be used for dynamic signalling in a large-network to avoid congestion mechanisms itself. The technical novelty is by looking at the run-time analysis which is portrayed as a graph where for even graphs as large as 25,000 nodes and 35,59,000 edges the run time is 1150s**. These run-times are reasonable in real-world applications and can be compared to any other implementation which provides such flow analysis results.

7. ARCHITECTURE

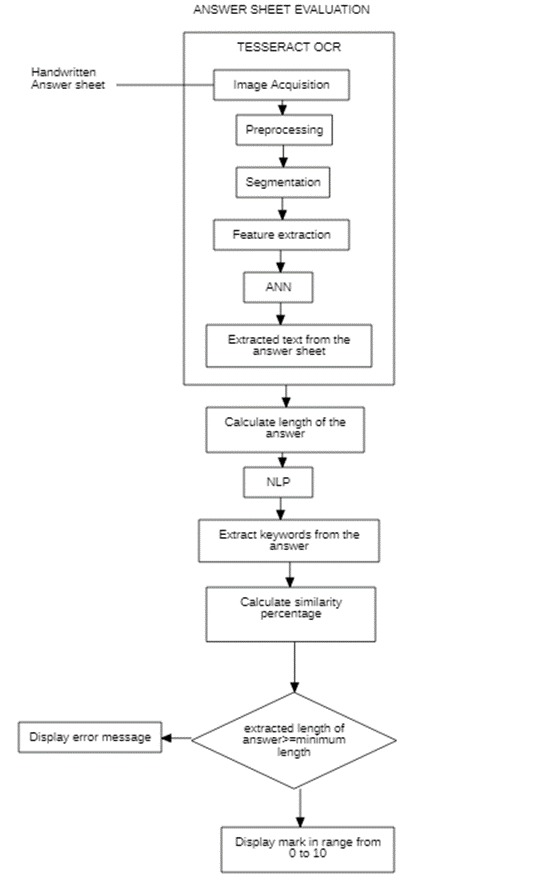
## 7.2 ABSTRACT ARCHITECTURE

## 

## 7.3EXPLANATION OF THE ARCHITECTURE

The system initially consists of a method that accepts the scanned image of the answer as images. This is then passed as input for the extraction of the words and sentences from the image of the answers. Thereafter the pre-processing of the text takes place after which keyword comparison is done. Here is where the pattern recognition happens and similarity checking is done. Finally, the score prediction is done based of the similarity and patterns found.

7.3 TASK ARCHITECTURE



## 7.4 EXPLANATION OF THE ARCHITECTURE DIAGRAM

The answer sheet image is uploaded as input. Using Python-tesseract, an optical character recognition (OCR) tool for Python and an open-source text recognition engine, text is extracted from images. Pytesseract , a wrapper for the Tesseract-OCR Engine is installed. We convert from image to string using the method image\_to\_string().we first converted the image to gray scale and then specified the kernel shape and size. Next, we found the contours and looped over them, chopping the rectangle area. Next, we have passed the rectangle area onto pytesseract for extracting text from it and then writing it into the text file.

Text file is pre-processed, where Sentence Segmentation, Tokenization, Removing Stop Word, and Word Stemming is done. Tokens are generated after pre-processing. The sequential pattern is generated from the tokens of each sentence. The count for each obtained pattern is also determined. The cosine-based similarity approach is used to calculate the similarity between the input answer and the model description. To compute cosine similarity between two sentences sentence number one and sentence number two, the sentences are turned into terms or words, words are transformed in vectors. The given answers are allotted a score by identifying both commonalities and similarity between the students’ answers. Finally, scores are printed.

8. DETAILED MODULE DESIGN

## 8.1 IMage recognition MODULE

The scanned images of the answer are first uploaded in sequence. The text from the image is extracted using a Python programming-based library called pytesseract. To read the text from an image the first step is to open the image. This can be done via the open() method from the Image object of the Pillow library. Next, to actually read the text from an image, the image object that has been opened must be passed to the image\_to\_string() method of the Pytesseract module. The image\_to\_string() method converts the image text into a Python string which can now be used for further pre-preprocessing.

## 8.2 PREPROCESSING Module

The input text is then pre- processed and changed into a normalized form. There are four main activities that are performed in pre-processing.

* Segmentation
* Tokenization
* Removal of Stop Words
* Word Stemming

Sentence segmentation involves splitting the paragraphs present into sentences. Tokenization involves reduces them to words. Next is the removal of stop words. Stop words are a set of commonly used words in a language. Examples of stop words in English are **“a”, “the”, “is”, “are”** and etc. These have to be removed as they add no meaning to the answer that has been written. The last step is stemming which removes the prefixes and suffixes of the word and produces its root form which is essential to us for key word checking.

## 8.3 PATTERN MINING MODULE

Tokens are generated after pre-processing. The sequential pattern is generated from the tokens of each sentence. The count for each obtained pattern is also noted. Most frequently observed patterns are also obtained based on the count.

## 8.4 SIMILARITY COMPARISON MODULE

Cosine similarity is one of the metrics to measure the text-similarity between two documents irrespective of their size in Natural language Processing. A word is represented into a vector form. The text documents are represented in n-dimensional vector space.

Mathematically, Cosine similarity metric measures the cosine of the angle between two n-dimensional vectors projected in a multi-dimensional space. The Cosine similarity of two documents will range from **0 to 1**. If the Cosine similarity score is **1**, it means two vectors have the same orientation. The value closer to 0 indicates that the two documents have less similarity.

The mathematical equation of Cosine similarity between two non-zero vectors is:



## 8.5 EVALUATION MODULE

The score is thereby allotted once the similarity is checked and the commonalities between answers have been considered. A score out of 10 is assigned to an answer and is marked for a student.

9. IMPLEMENTATION

## 9.1 IniTIAL Set-up:

The set up involves the running environment for the code. System requirements would involve the software constraints of an operating system capable of running the latest version of Python (here, Windows 10) and hardware constraints of at least 1 GB of memory.

## 9.3 Tools:

Jupyter Notebook, JupyterLab

## 9.4 Code Snippets:

### 9.4.1 IMAGE PROCESSING MODULE

### When given an image as input

### 

### Results in

### 

### 9.4.2 PRE-PROCESSING MODULE

### 

### 

### 9.4.2 PATTERN MINING AND SIMILARITY COMPARISON MODULE

### 

### 

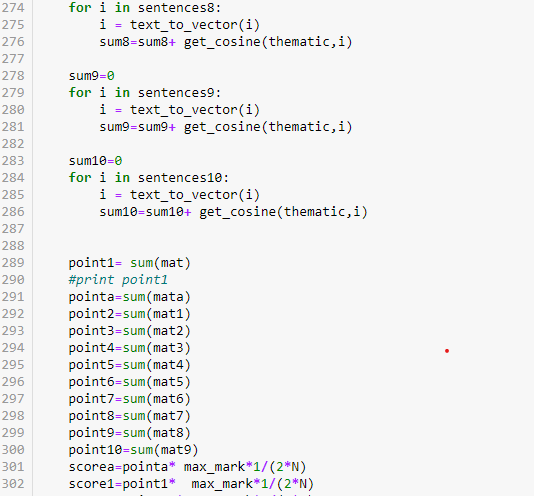
### 

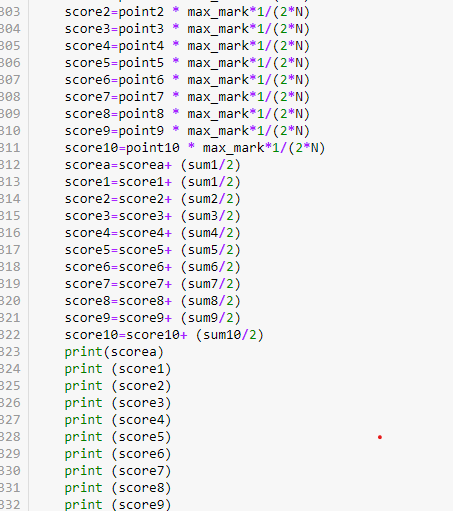
### 9.4.3 EVALUATION MODULE

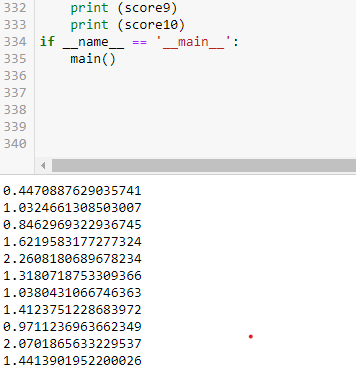
### 

### 

### 







10. SUPPORT INFORMATION

11. SURVERYED CONTENT

12. RESULTS AND COMPARISON

13. CONCLUSION AND FUTURE ENHANCEMENTS

14. List of References

15.Appendix A:

The undersigned acknowledge they have completed implementing the project “Traffic Analysis in large network graphs in spark” and agree with the approach it presents.

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: | 04/04/2016 |
| Name: | Sanjana Jain S |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: | 04/04/2016 |
| Name: | Shivanee N |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: | 04/04/2016 |
| Name: | Varun R |  |  |

16. Appendix B: References

The following table summarizes the research papers referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name and Version** | **Description** | **Location** |
| A mapreduce-based maximum-flow algorithm for large small-world network graphs | Hadoop based max-flow analysis and results of various stages and rounds. | ieeexplore.ieee.org/xpls/abs\_all.jsp?arnumber=5961676 |
| An internet traffic analysis method with mapreduce | Traffic is analysed based on various features using mapreduce stages. | ieeexplore.ieee.org/xpls/abs\_all.jsp?arnumber=5486551 |
| Network information flow | Work indicating on how information would flow in a large network. | people.csail.mit.edu/arasala/papers/networkFlow |
| Pregel: Large Graph Processing System by Google | The first of a kind, graph processing system which works at a higher efficiency when compared with mapreduce. | https://kowshik.github.io/JPregel/pregel\_paper |

17.Appendix C: Key Terms

The following table provides definitions for terms relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Big Data | **Big data** is a term for **data** sets that are **large** or complex that traditional **data** processing applications are inadequate |
| RDD | Resilient Distributed Database is a format used for storage in Spark |
| Maximum Flow | Given a network with A **flow** network, with source s and sink t, maximum flow is the largest value of units that pass across without any overflow. |
| LogNormal | Logarithm of the Normal Curve is taken (bell-shaped curve). |
| Augmenting Path | An **augmenting path** is a simple **path** - a **path** that does not contain cycles - through the graph using only edges with positive capacity from the source to the sink. |