Software Requirements Specification

For

Scoring algorithm for short answers and essays

Version 1.0

Prepared by

Sushma N	: 1PE12IS106
Steffi Crasta	: 1PE12IS414
P Karan Jain	: 1PE12IS065

Guided by

Dr.Gowri Srinivasa

Table of Contents

Chapter 1. INTRODUCTION

1.1 Introduction

Essays and short answers are crucial testing tools for assesing academic achievement, integration of ideas and ability to recall, but are expensive and time consuming to grade manually. Manual grading of essays takes up a significant amount of instructors' valuable time, and hence is an expensive process. Automated grading, if proven to match or exceed the reliabilty of human graders, will significantly reduce costs.

1.1.1 Purpose of the project

One of the key roadblocks to teaching and evaluating critical thinking and analytical skills is the expense associated with scoring tests to measure those abilities. For example, tests that require “constructed responses” (i.e., written answers, written essays) are useful tools, but they typically are hand scored, commanding considerable time and expense from public agencies. So, because of those costs, standardized examinations have increasingly been limited to using “bubble tests” that deny us opportunities to challenge our students with more sophisticated measures of ability.

1.1.2 Scope

The product has scope in departments of education where developing new forms of testing and grading methods, to assess the new common core standards. For

example, we know that essays are an important expression of academic achievement, but they are expensive and time consuming for states to grade them by hand. So, we are frequently limited to multiple-choice standardized tests. We believe that automated scoring systems can yield fast, effective and affordable solutions that would allow states to introduce essays and other sophisticated testing tools.

Benifits:

- Human time and effort is saved.

- Coherence in evaluation of all the scripts present

Objective:

- To evaluate and assign a score for a short answer/essay without human intervention .

- To provide visualization of a scores for group of students.

1.2 Literature survey

The automatic evaluation of short answers has been tried by various people around the world. It was also a Data Science competition hosted by Kaggle sponsored by The Hewlett Foundation, the best results obtained were by Tandalla (2012)'s approach, the best performing one of the competition, achieved a Quadratic Weighted (QW) Kappa of 0.70 using just regular expressions as features. Tandalla (2012)'s was the best performing model at the ASAP-Short Answer Scoring competition. One of the important aspects of Tandalla's approach was the use of manually coded regular expressions to determine whether a short answer

matches (or does not match) a sample pattern. Specific regular expressions were developed for each prompt set, depending on the type of answers each set evoked (e.g.presence of words such as “alligator”, “generalist”,“specialist” etc. in the text). These patterns were entirely hand-coded, which involved a lot of manual effort. Tandalla built a Random Forest model with the regular expressions as features. This model alone achieved a QW Kappa of 0.70. Tandalla alsomanually labeled answers to indicate match with the rubric text. A detailed description of the best performing approach is available in Tandalla (2012).

However, regular expression generation can be tedious and time consuming, and the performance of these features is constrained by the ability of humans to generate good regular expressions. Automating this approach would ensure that the process is repeatable, and the results consistent.

Leacock and Chodorow (2003) developed the use of a short-answer scoring system called C-rater, which focuses on semantic information in the text. They used a paraphrase-recognition based approach to score answers.

Bachman et al. (2002) proposed the use of a short answer assessment system called WebLAS. They extracted regular expressions from a model answer to generate the scoring key.

Implementations of learning algorithms have been applied like finding out graph based alignments and lexical similarities, this way patterns could be generated are automated.

For Essay scoring algorithms also a competition was held by Kaggle sponsored by The Hewlett Foundation where 8 sets of essays in ASCII written by students from grade 7 to grade 10 are provided as data set. Features

used were fluency and dexterity, diction and vocablury, structure and organization, Orthography and content. The model used for learning and evaluating was logistic regression and a average Quadratic weighed kappa score obtained was 0.73 across all the sets.

Alen Lukic and Victor Acuna used and compared two methods for predicting scores for the essays in the validation set. The first was a simple naive Bayes prediction, which assumes that the scores are independent of each other. The model is trained on the features of the training essays and predicts scores for the validation essays given their features.The second method used was the closed-form solution to linear regression.

The naive Bayes predictor slightly outperformed the closed-form model in the benchmarktest. The quality of the naive Bayes predictions decreased with the addition of features other than word count, whereas that of the closed-form model increased. The closed-form model slightly outperformed the naive Bayes model with the addition of the remaining features. The results were in line with the hypothesis. For the closed-form model, the κ value was approximately 0.0363 higher than hypothesized. However, the closed-form model didn't outperform the naive Bayes model as distinctly as predicted. The goodness of this model compared to simple naive Bayes prediction for the problem of automatic essay grading was inconclusive.

However the authors have suggested that logistic model
trees, ngram , k nearest neighbours for bag of words would all improve the System. Another possibility for improvement is the extraction of more complex

essay features. For example, the noun-verb relatedness graphs which we constructed may have been too local, as they only connected nouns and verbs if they were the respective adjacent parts of speech to each other in a sentence.

1.2.1 References :

[1] Manvi Mahana, Mishel Johns, Ashwin Apte. (2012). Automated Essay Grading using Machine Learning, Stanford University.

[2] Likic, A., & Acuna, V.(n.d.). Automated Essay Scoring, Rice University.

[3] Lakshmi Ramachandran, Jian Cheng & Peter Foltz "Identifying Patterns For Short Answer Scoring Using Graph-based Lexico-Semantic Text Matching"

1.3 Existing System

The Existing system for short answers is either through Optical mark recognition answer sheets where the answers to questions are options provided or only a particular key word is searched for in the short answer with the help of regular expressions. For essays still human correction is needed, however ETS (Educational Testing Service) offers e-rater®, an automated essay scoring program. It was first used commercially in February 1999. Jill Burstein was the team leader in its development. ETS's Criterion^{SM} Online Writing Evaluation Service uses the e-rater engine to provide both scores and targeted feedback.

1.4 Proposed System

In order to overcome the problems faced in the above system natural processing of text and using Graph-based Lexico-Semantic Text Matching a score would be

generated for the short answer. The system will detect if the given context is short answer or an essay with the h Automated evaluation of short answers and It generates instantaneous result and correction is done syntactically as well as Semantically. For Essays simple and complex features would be extracted using various models and N-gram model would be used for generating the scores for the Essay.

1.5 Problem Statement

Automated evaluation of short answers and essay is done using algorithm to correct syntactically, semantically and assign scores. This reduces high cost and the slow turnaround of hand scoring thousands of written responses in standardized tests. We have to extract several features of the essays, ranging from simple numerical data such as word count and average word length, content-specific numerical data such as misspelled word count and adjective count and we look for specific response in short answers. The system would assign preliminary grades to all student essays, and the instructors would only become involved in the process to address student disputes. Automated grading, if proven to match or exceed the reliability of human graders, will significantly reduce costs. We have to implement and train machine learning algorithms to automatically assess and grade essay responses. These grades from the automatic grading system should match the human grades consistently. Finally we need to analyse scores of set of students.

1.6 Summary

This chapter describes the idea of project in brief. It begins with an explanation of

the purpose of this project and scope of the project. Later literature survey is described in detail and also brief about existing system and proposed system. Finally we explain the problem in hand that is being designed.

Chapter 2. Software Requirement Specifications

2.1 Software Requirement Specification

Functional Requirements :

• Accept the answers as the input for the question provided.

• Pass it through the system and obtain a computer evaluated scorecard.

• Evaluate the scores for a set of students in the class.

• Provide visualization and analytics for specified range of students/questions.

Non Functional Requirements :

Non-functional requirements pertain to other information needed to produce the correct system. These requirements are not functional in nature, these are the constraints within which the system must work.

Performance: The system can take longer time to process the answer script but the grades should be consitent with the scores provided by human evaluation.

Scalability: The system should be scalable enough to store all the attributes related to the short answers and the essay.

Usability: User interaction has to be made comfortable. For each answer or an essay a score card would be generated.

Reliability: The system would provide reliable answers, however human intervention would be needed in high stake applications.

Flexibility: System can be used for evaluation of short answers and essays in any exam.

2.2 Operating Environment

2.2.1 Hardware Requirements

Processor: Intel core i3

Memory: 256 MB

Hard disk : 2 GB

2.2.2 Software Requirements

The application shall operate on a running the current corporate approved versions of:

- Python 2.7.x or Python 3
- Textmining package for Python.
- Natural Language Toolkit (NLTK) package present for Python
- Scikit-learn package for Python

2.3 Background modules

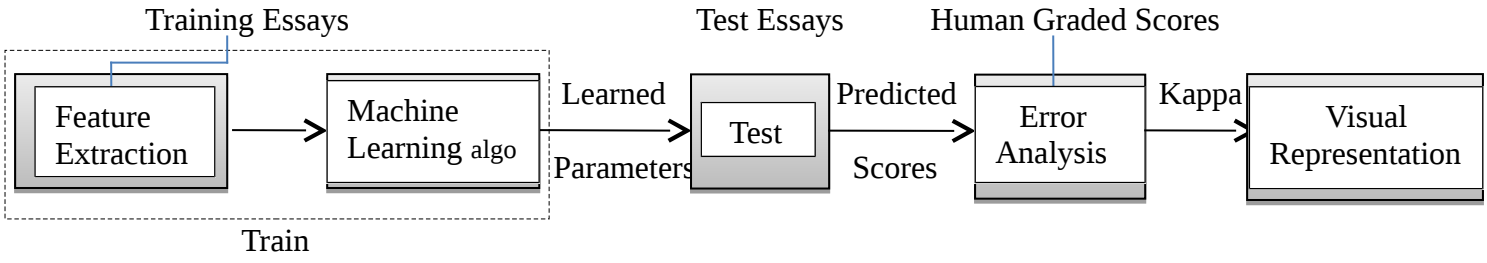
Various modules have been used to implement the solution to this problem.

- **NLTK**: The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for the Python programming language.
- **POS Tagger**: A part-of-speech tagger, or POS-tagger, processes a sequence

of words, and attaches a part of speech tag to each word.

- **Tokenization**: A tokenizer that divides a string into substrings by splitting on the specified string.
- **PyEnchant:** is a spellchecking library for Python, based on the excellent Enchant library.
- **Wordnet:** is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.

2.4 Architecture



Layered Architectural Model

The above system represents the layered architecture of the system. The various levels of the architecture are explained below

Training Input : The system takes various the input training set of essays and extracts features from the essays. The features are then passed to various machine learning algorithms for the classifier to learn and generate patterns and this way the classifier is trained.

Input : The essay to be evaluated is then passed to the classifier.

Output: The evaluated essay is given a score and the score is compared to the human graded score. The error in the scores is calculated and error rate is called Kappa.

Visual Representation : User could pass some queries to generate reports for generating visuals of various students or questions depending on the quesries passed by the user.