AUTOMATED ESSAY GRADING USING DEEP LEARNING

A Project Report submitted to

Jawaharlal Nehru Technological University, Hyderabad

In partial fulfilment for the requirement for the award of Bachelor of technology Degree in

Computer Science and Engineering

Submitted By

Kotha Nithya	**17UK1A0544**
Poshala Amulya	**17UK1A0502**
Sharath Kumar Gongalla	**17UK1A0557**
Podila Madhu	**17UK1A0542**

Under the Guidance of

Dr. Rakesh Nayak

(Professor, CSE Dept.)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliation to JNTU, Hyderabad & Approved by AICTE, New Delhi.

Bollikunta, Warangal (T.S)-506005

2017-2021

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Warangal

CERTIFICATE

This is to certify that the project report entitled "***AUTOMATED ESSAY GRADING USING DEEP LEARNING***" is submitted by ***Kotha Nithya*** (17UK1A0544),***Poshala Amulya*** (17UK1A0502),***Sharath Kumar Gongalla*** (17UK1A0557),***Podila Madhu*** (17UK1A0542), in partial fulfilment of the requirements for the award of the Degree in Bachelor of Technology in computer science and engineering during the academic year 2020-2021.

Guide:

Dr. Rakesh Nayak

Professor

HOD:

Dr. R. Naveen Kumar

Professor

External Examiner:

ACKNOWLEDGEMENT

This project has been carried out in the department of Computer Science and Engineering of Vaagdevi Engineering College, Bollikunta, Warangal. Many people have helped us in the realization of this work we would like this opportunity to express our gratitude to all of them.

We express our gratitude to our principal **Dr. P. Prasad Rao**, who permitted us to carry the project work as part of the academics.

We would like to thank **Dr. R. Naveen Kumar**, Head of Department (CSE) for his support and encouragement in completing our project. We would like to express our gratitude to our Project guide **Dr. Rakesh Nayak** Professor for his support and encouragement in completing our mini project successfully.

We express our sincere thanks and gratitude to **TheSmartBridge**, for providing internship, for constant support and giving necessary guidance for completion of our major project.

Finally, we wish to take this opportunity to express our deep gratitude to our family members and all the people who have extended their cooperation in various ways during our project work.

ABSTRACT

Essays are crucial testing tools for assessing academic achievement, integration of ideas and ability to recall, but are expensive and time-consuming to grade manually. Manual grading of essays takes up an amount of instructors' valuable time and hence is an expensive process. In the view of educational institutions, assignments or essays play an important role in assessing the ability of students to understand and recall the topics explained to the students. The manual evaluation of these papers takes a lot of effort and time of the evaluators hence resulting in a time consuming process.

The solution to grade a large number of papers effectively within a stipulated time is to let the machine do the grading. The automated grading system will not only reduce the time of evaluation but comparing it with human scores will also make the score more realistic.

Automated grading, if proven to match or exceed the reliability of human graders, will reduce costs. Currently, automated grading is used instead of second graders in some high-stakes applications, and as the only grading scheme in low stakes evaluation. This application can have a high utility in many places. For instance, currently, evaluation of essay writing sections in exams like GRE, GMAT, and TOEFL is done manually. And, so automating such a system may prove to be highly useful.

An automated grading system is built with the magical powers of neural networks. Using automation reduces time and effort in evaluation. NLTK libraries for feature extraction and LSTM are used for the learning process.

The project aim is to develop a system which grades an essay or a paper without any manual involvement. When an essay is loaded into the proposed grading system, the system accepts the essay given as the input and grades it using deep learning techniques and its layers such as LSTM and dense layers.

Keywords: Grading, evaluation, score, deep learning, layers, educational institutions

LIST OF CONTENTS

LIST OF FIGURES

LIST OF SCREENS

CHAPTER-1
INTRODUCTION

Nowadays most educational institutes are moving to internet-based methods to perform exams. Especially, the English Language Examination which is taken by an enormous number of people around the world for many purposed. Automating the grading will be so beneficial as the exam takers do not have to wait for at least two weeks to get the results, and the educational institutes will save money and effort simply by enabling the machine-grader program which has been trained and designed to achieve the highest performance.

Essays are a tool for testing the students' fluency, vocabulary and grammatical correctness in a language. They are also useful to test one's creativity, originality and articulateness. The highly subjective and diverse nature of an individual in writing an essay makes it difficult to grade the essay uniformly across many human graders. In addition to this there are various other biasing factors in grading an essay. There has been research on automatic essay grading since the 1960s. The first systems based their grading on the surface information from essays. These systems were successful though they failed to capture aspects like grammatical correctness and language fluency. Much research has been conducted in the field most notably by Educational Testing Service (ETS). Clubbing this together with the resurgence in new technologies such as neural networks, deep neural networks, there is a whole new world of possibilities due to their capacity of modelling complex patterns in data. These methods do not depend on feature engineering so they are really useful for solving problems in an end-to-end fashion. With this intuition this project aims to the relevant knowledge in the field of education and try to create an essay grader which can make quality education more accessible. The work also explores methods of improving the quality and usability of the system.

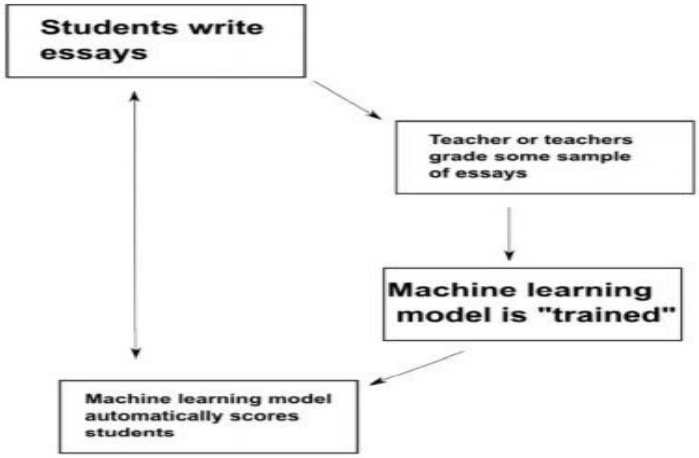


Fig 1.1 Process of Automated Essay Grading

In today's education system, essays and literary pieces form an integral part of assessing one's academic understanding, ability to integrate and express ideas meaningfully. It is observed that grading such written pieces is a time-taking process and take up a huge chunk of instructor's time. Also, cases of manual grading sometimes being non-transparent are not unheard of. So, we plan to tackle this challenge by introducing a method to automatically grade written pieces. Automated essay scoring is a measurement technology in which computers evaluate written work . With a model that can grade with good accuracy, a lot of instructor's time can be saved and an impartial scoring for students can be achieved.

Essays play a vital role in trying out or assessing instructional achievement, integration of ideas and capacity to take into account of a student. If a student has better understanding of a concept, then he or she can apply the concept in a better way compared to those who don't have a clearer understanding of the concept. Grading students' assignments creates awareness about his or her learning ability, understanding of concepts and their application to the teachers. If the time taken for evaluation of these essays is somewhat reduced, then teachers can give more attention to preparing more notes for the next classes, gather up more conceptual examples for the better understanding of the students.

Efforts to extend standardized tests beyond multiple choice questions are limited by the ability to grade responses. Potential applications of machine learning and natural language processing could allow for these methods to scale to free text. A consortium of 44 US States and the Hewlett Foundation are looking into systems to automatically grade standardized test essays using small amounts of manually labelled training examples.

Analyzing natural language, or free-form text used in everyday human-to-human communications, is a vast and complex problem for computers regardless of the medium chosen, be it verbal communications, writing, or reading. Ambiguities in language and the lack of one "correct" solution to any given communication task make grading, evaluating or scoring a challenging undertaking. In general, this is a perfect domain for the application of machine learning techniques with large feature spaces, and huge amounts of data containing interesting patterns.

These grades from the automatic grading system should match the human grades consistently. Currently, automated grading is used instead of second graders in some high-stakes applications, and as the only grading scheme in low stakes evaluation. We make use of Supervised Learning methods on the human-scored "labelled" essays. As a learning model, RNN seems to work very well in similar tasks.

Our project aims at developing a model using deep learning techniques which automatically grades an essay upon submission. We grade our essay on a scale of 1-10 and the result would be displayed on the screen.

1.1 Motivation

Automated Essay Grading is a tool for evaluating and scoring of essays written in response to specific prompts. It can be defined as the process of scoring written essays using computer programs. The process of automating the assessment process could be useful for both educators and learners since it encourages the iterative improvements of students' writings.

Automated grading if proven effective will not only reduce the time for assessment but comparing it with human scores will also make the score realistic.

The human graders unknowingly tend to grade an essay biasing towards the individual subject matter presented. Another major drawback is the time required to grade essays can be significantly high. The present technologies present an excellent opportunity to automate tedious tasks such as essay grading. Availability of powerful Deep Learning libraries is a major push towards the reliance and devising the system. This project will help in maintaining an unbiased and fair approach towards evaluating the written essay for competitive exams, tests, etc. which is in turn beneficial in multiple ways.

1.2 Problem definition

Analyzing natural language, or free-form text used in everyday human-to-human communications, is a vast and complex problem for computer machines irrespective of the medium that has been selected, be it in the form of verbal interaction, written sample, or reading. The murkiness in the given system language and the lack of one definite output to any given communication task make grading, evaluating or scoring a difficult challenge. In general, in this particular domain implementing machine learning techniques with various features spaces, and large collection of data having different pattern can give us better outcomes.

Our aim is to build a model that can take in an essay and automatically outputs the grade of that essay. Our model is capable of acknowledging the difference in scale and outputting the corresponding grade. So here the main goal is to build a model that outputs the grade of a given essay.

1.3 Objective of Project

The reason for the lack of reliability in some of the automated essay scoring is that they use very basic features like word count, paragraph count, and sentence length. This causes automated essay

scoring systems to focus more on the size and the structure of the essay rather than the content and quality of the essay. One positive development in the field of automated essay evaluation is the growing amount of data available to work with, which makes machine learning an attractive option to solve this problem. However, a grading model must learn from data that represents a noisy relationship between essay attributes and its grade. The objective of this project is to combine quantitative features with essay content to improve the reliability of the automated essay grading.

Our project aims at developing a model using deep learning techniques which automatically grades an essay upon submission. We grade our essay on a scale of 1-10 and the result would be displayed on the screen.

1.4Limitations of Project

- Need high memory bandwidth.
- Prone to overfitting.
- In complex query language, the system may not be able to provide the correct answer it a question that is poorly worded or ambiguous.
- Slow and Complex training procedure.
- Limited to English essays.

1.5Project Synopsis

The data set has various factors such as ‘essay_id’, ‘essay_set’, ‘essay’, ‘domain1_score’, etc. The deep learning techniques like natural language processing, LSTM are used in order to predict the outcome of the attribute – “domain1_score”.

CHAPTER-2

LITERATURE SURVEY

2.1 INTRODUCTION

In recent decades, large-scale English language proficiency testing and testing research have seen an increased interest in constructed-response essay-writing items. The TOEFL iBT, for example, includes two constructed-response writing tasks, one of which is an integrative task requiring the test-taker to write in response to information delivered both aurally and in written form (Educational Testing Service, n.d.). Similarly, the IELTS academic test requires test-takers to write in response to a question that relates to a chart or graph that the test-taker must read and interpret (International English Language Testing System, n.d.). Theoretical justification for the use of such integrative, constructed-response tasks (i.e., tasks which require the test-taker to draw upon information received through several modalities in support of a communicative function) date back to at least the early 1960's. Carroll argued that tests which measure linguistic knowledge alone fail to predict the knowledge and abilities that score users are most likely to be interested in, i.e., prediction of actual use of language knowledge for communicative purposes in specific contexts:

An ideal English language proficiency test should make it possible to differentiate, to the greatest possible extent, levels of performance in those dimensions of performance which are relevant to the kinds of situations in which the examinees will find themselves after being selected on the basis of the test. The validity of the test can be established not solely on the basis of whether it appears to involve a good sample of English language but more on the basis of whether it predicts success in the learning tasks and social situations to which the examinees will be exposed.

Moreover, Canale and Swain (1980), in their elaboration on the notion of communicative competence, explained that demonstration of linguistic mastery is a necessary but not sufficient condition for inferring communicative language ability on the part of an individual, and that students of foreign languages must be given opportunities "to respond to genuine communicative needs in realistic second language situations … not only with respect to classroom activities, but to testing as well" .

Bachman and Palmer (1996) echoed the concern that most language testing purposes are such that test constructs should define communicative language ability and not simply linguistic knowledge. In their discussion of the qualities of language tests that enhance test usefulness, they included authenticity of test tasks as a major contributing element. Authenticity in this context is defined as "the degree of correspondence of the characteristics of a given language test task to the features of a TLU task". If test tasks closely resemble the real-world language use situations that are of interest to stake-holders, then it is assumed that this will enhance the predictive power of the tasks, and, therefore, the overall validity and usefulness of the test.

When the target language use domain of1` interest to test-developers is that of the writing activities of university students, it has often been seen as a prima facie minimum requirement for authenticity that the test tasks involve the production of an actual written response that requires examinees to integrate knowledge in service of a communicative task in the way that might be called for when writing an academic essay. Popham (1978) does not mince words on the topic:

As a measure of certain kinds of complex learning outcomes, the essay item is unchallenged. Since this kind of item sometimes requires a student to put together ideas and express them in original ways, there is no way of simulating that kind of requirement in a selected-response item or, for that matter, even in a short-answer item.

However, constructed-response writing tasks have both advantages and disadvantages. Unlike multiple-choice items which have a single criterion for correctness, experts often disagree on how to operationalize and score the set of qualities that define excellent writing. While the fact that constructed-response essay items require students to generate samples of normative language (rather than simply selecting them) may make such items a more proximal measure of communicative writing ability, the process of scoring essay items is quite complex. Human raters must be hired and trained to score each of the examinee essays. The additional time required for this process means that test scores cannot be reported to examinees as quickly as would be possible for machine-scored multiple-choice items (Livingston, 2009). Moreover, the costs associated with this process are passed on to the examinees themselves in the form of testing fees. In addition, the use of human raters introduces a new challenge to maintaining the reliability and construct validity of test scores, as raters are bound to differ in their perceptions of candidate performances and their tendencies towards leniency and severity. Raters may also have unconscious biases that are not immediately amenable to correction through training.

Perhaps it is for the above reasons that the testing industry was initially slow to include constructed-response tasks on large-scale, high-stakes assessments. It was not until 1986 that Educational Testing Service (ETS) began to offer the Test of Written English, the first performance-based examination of second language writing ability to be offered by the organization. Prior to this time, writing ability had been measured indirectly on the TOEFL with multiple-choice items focusing on "knowledge of structure and written expression". Also for the reasons mentioned above, the advent of computer technology has led to multiple attempts to create automatic scoring applications that might allow for cheaper, more reliable, and in the view of some proponents, even more accurate scoring of test-taker performances. The pages that follow will provide a discussion of some of the challenges associated with the performance-based assessment of writing as well as a critical review of the automated essay scoring applications that have been developed as an alternative to or supplement for human ratings of writing performance.

2.1.1 Challenges Associated with Essay Grading:

A basic requirement of ethical language testing is that test developers provide evidence in the form of research findings for the interpretive argument that underlies specific claims regarding the inferences that may be justified by test scores. When assessing writing ability on the basis of a test taker's written performance, construct-irrelevant and unreliable variance in the ratings are a chief concern. Such variance may occur when raters differ in their perceptions and preconceived notions regarding good writing and good writers. If individual raters lack the ability to give consistent ratings (intra-rater reliability) and to rate in a normative way (i.e., in the way that is intended by the test designer, consistent with the descriptions in the rating scale, and in agreement with other raters [inter-rater reliability]), the validity of inferences based on the scores of writing performance assessments will be diminished. In order to maximize the intra-rater and inter-rater reliability of ratings given by human raters, the usual practice is to have at least two raters assign either holistic or component scores to written performances with the aid of a rating scale and following some kind of rater training. In actual practice, however, rater training often consists of a single norming session in which raters practice giving ratings and compare their own ratings to those of other raters . Achieving consistency of rating and normative rating behaviors on the part of essay raters requires a significant investment on the part of both testing programs and the raters themselves. Testing programs must train raters and pay them for their time, while the raters themselves must spend long hours rating multiple essays at a sitting. Kim (2010) found that repeated rater training and feedback sessions were necessary to achieve internal consistency and normative rating behavior on the part of even experienced raters. Even large testing companies may struggle to allocate the resources necessary to ensure that raters receive adequate training. Several studies have highlighted the challenge of assigning reliable and construct-valid scores when using human raters to score writing performances.

Shohamy et al. (1992) examined intra-rater and inter-rater reliability for a diverse group of raters who were divided into a group that received rater training and a group that did not. An encouraging finding of the study was that, compared to rater background, rater training had a much larger and positive effect on both intra-rater and inter-rater reliability. However, the authors note that intensive, repeated rater training including ample opportunity for discussion and feedback is needed to achieve the most desirable levels of reliability. Training of this type may not be feasible for large testing companies employing thousands of raters in remote locations. Kondo-Brown's (2002) study is a good illustration of a different challenge to the reliability of human ratings. The study examined the severity of ratings given to Japanese L2 writing performances by a group of raters who were homogeneous with respect to background and native language. The raters in the study were all native speakers of Japanese, taught in the same university, and held advanced degrees in language-related

fields. While high inter-rater correlation coefficients were observed for the three raters in the study, the raters nevertheless displayed significant individual differences in severity with respect to particular aspects of writing performances. In particular, significantly biased ratings were more common for candidates with very high or low ability, suggesting that despite rater training the raters were not able to be consistent in their application of rating criteria to examinees at all ability levels. Kondo-Brown concludes that while rater training may improve inter-rater reliability and the internal consistency of ratings, it may not have much impact on other undesirable rater characteristics such as increased or decreased severity when rating certain examinees, items, or for certain areas of the rating scale.

Similarly, Shi (2001) examined holistic ratings given by native and non-native English speaking raters of EFL writing in a Chinese university and found that, while the two groups did not differ significantly in the scores they gave, the raters differed greatly in the justifications that they gave for their ratings. Self-report data suggested that different raters weighted different essay characteristics more heavily while arriving at a holistic score. Native speakers tended to focus more on content and language while non-native speakers focused on organization and length. Therefore, although they may have given similar scores to an essay, the raters in Shi's study gave these scores for very different reasons, suggesting that they did not share a common understanding of the test construct. Shi notes that if students were to receive feedback from raters such as the ones in her study, they would likely be confused by contradictory messages. While it is often assumed in the field of language assessment that human ratings, once reliable, are inherently valid, Shi concluded that the findings underline the lack of a one-to-one correspondence between the reliability and construct-validity of human ratings and shed light upon the need for the development of rating procedures that promote more construct-valid ratings, such as the use of analytic rubrics that encourage more thorough and balanced attention to the construct.

2.2 EXISTING SYSTEMS

2.2.1 Project Essay Grade

Ellis Page developed the PEG in 1966. PEG is considered the earliest AES system that has been built in this field. It utilizes correlation coefficients to predict the intrinsic quality of the text. It uses the terms ‘‘trins’’ and ‘‘proxes’’ to assign a score. Whereas ‘‘trins’’ refers to intrinsic variables like diction, fluency, punctuation, and grammar, ‘‘proxes’’ refers to correlations between intrinsic variables such as average length of words in a text, and/or text length.

The PEG uses a simple scoring methodology that consists of two stages. The former is the training stage and the latter is the scoring stage. PEG should be trained on a sample of essays from 100 to 400 essays, the output of the training stage is a set of coefficients (β weights) for the proxy

variables from the regression equation. In the scoring stage, proxes are identified for each essay, and are inserted into the prediction equation. To end, a score is determined by estimating coefficients (β weights) from the training stage.

Some issues have been marked as a criticism for the PEG such as disregarding the semantic side of essays, focusing on surface structures, and not working effectively in case of receiving student responses directly (which might ignore writing errors). PEG has a modified version released in 1990, which focuses on grammar checking with a correlation between human assessors and the system ($r = 0.87$).

Measurement Inc. acquired the rights of PEG in 2002 and continued to develop it. The modified PEG analyzes the training essays and calculates more than 500 features that reflect intrinsic characteristics of writing, such as fluency, diction, grammar, and construction. Once the features have been calculated, the PEG uses them to build statistical and linguistic models for the accurate prediction of essay scores.

2.2.2 E-rater

Educational Testing Services (ETS) developed E-rater in 1998 to estimate the quality of essays in various assessments. It relies on using a combination of statistical and NLP techniques to extract linguistic features (such as grammar, usage, mechanics, development) from text to start processing, then compares scores with human graded essays.

The E-rater system is upgraded annually. The current version uses 11 features divided into two areas: writing quality (grammar, usage, mechanics, style, organization, development, word choice, average word length, proper prepositions, and collocation usage), and content or use of prompt-specific vocabulary.

The E-rater scoring model consists of two stages: the model of the training stage, and the model of the evaluation stage. Human scores are used for training and evaluating the E-rater scoring models. The quality of the E-rater models and its effective functioning in an operational environment depend on the nature and quality of the training and evaluation data. The correlation between human assessors and the system ranged from 0.87 to 0.94.

2.2.3 Latent Semantic Analysis

An alternative to what Ben-Simon and Bennett refer to as the “brute empirical” (surface feature) approach to automatic essay scoring applied in PEG and E-Rater is the latent semantic analysis-based approach (LSA). LSA is described by its developers as “a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations

applied to a large corpus of text". LSA was developed not only as a practical tool for applications, such as meaningful corpus searching and the scoring of essays, but also "as a model of the computational processes and representations underlying substantial portions of the acquisition and utilization of knowledge" . Unlike PEG, which focuses entirely on incidental surface features, or E-Rater, which combines incidental surface features with the more intuitive, such as vocabulary and grammatical accuracy, LSA ignores superficial surface features and word order entirely, attending instead to the statistical relationships in an examinee essay between meaningful units (i.e., words, sentences, paragraphs, and texts). Understanding the processes that underlie LSA is daunting for the mathematically uninitiated, but its basic premise is that the meaning of a word, sentence, or text, and the concepts embodied by each, are closely related to the conceptual contexts in which each occurs. Therefore, by measuring the collocations of meaningful units of text, a computer program can 'learn' what knowledge or concepts are contained in a piece of writing.

Foltz, Landauer, and Laham (1999) developed an LSA-based automatic essay scoring tool, Intelligent Essay Assessor (IEA), designed to attend to essay content rather than style. Unlike PEG and E-Rater, IEA trains not on a sample of essays scored by human raters, but on domain-representative text (e.g., actual essays written by college students or course textbooks). IEA, therefore, is used to determine the degree of congruence between an essay of known quality and a candidate essay. Since LSA represents semantic information as likely collocations between both words and similar words/groups of words in larger units of text (and not simply by word matching), Foltz et al. (1999) claim that it can recognize essays that are similar in content even if they differ markedly in vocabulary, grammar, and style. This means that IEA might be used to offer substantive feedback on the content of examinee responses. In order to avoid inaccurate scoring of "unique" essays that may not be appropriately evaluated by IEA's algorithm, such essays are flagged as anomalous and rated by a human. Currently, IEA will flag essays that it deems highly creative, off topic, in violation of standard formatting, or too similar to another known essay (and thus a possible case of plagiarism).

To evaluate the accuracy of IEA in an actual scoring situation, Foltz et al. scored a sample of over 600 opinion and argument GMAT essays. IEA achieved correlations of .86 with the human graders for both the opinion and argument essays while the ETS raters correlated with each other at .87 and .86 for the two opinion and argument essays, respectively. Foltz et al. also reported on other evaluations of IEA for different content areas including psychology, biology, and history at the middle school, high school, undergraduate, and graduate levels. In each case, IEA's observed reliability was comparable to inter-rater reliability and within generally accepted ranges for the testing purpose. Foltz et al. also reported that other LSA-based tools have scored comparably to

humans on content-based assessments, such as the synonym portion of the TOEFL and a multiple-choice introductory psychology exam.

As an additional, qualitative line of inquiry into IEA's usefulness, Foltz et al. (1999) recruited a university psycholinguistics course to use IEA for essay scoring over the course of two years. IEA was trained for this purpose on a sample of passages from the course textbook. To verify reliability before operational use of IEA, the researchers graded sample essays from previous semesters with the program and observed a correlation of .80 with the average human scores. Students in the course were able to submit essays online and receive instant estimated grades accompanied by feedback and suggestions for subtopics that may be missing. The students were encouraged to use IEA as a tool for revising their essays iteratively until they were satisfied that their essays were ready to be submitted to the professor for final grading. In a survey at the conclusion of the study, 98% of the students reported that they would definitely or probably use IEA if it were available for their other classes. Foltz et al. (1999) suggest that IEA has potential to impact assessment at all levels, including as a supplement or replacement for human raters in standardized testing, as an aid and objective check for classroom teachers, and as a content feedback device for students.

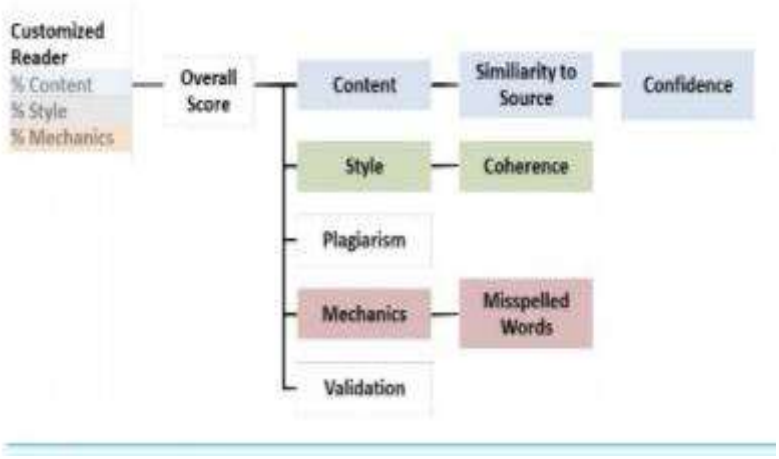


Fig 2.1 The IEA architecture.

2.2.4 IntelliMetric

Vantage Learning developed the IntelliMetric systems in 1998. It is considered the first AES system which relies on Artificial Intelligence (AI) to simulate the manual scoring process carried out by human-raters under the traditions of cognitive processing, computational linguistics, and classification.

IntelliMetric relies on using a combination of Artificial Intelligence (AI), Natural Language Processing (NLP) techniques, and statistical techniques. It uses CogniSearch and Quantum

Reasoning technologies that were designed to enable IntelliMetric to understand the natural language to support essay scoring.

IntelliMetric uses three steps to score essays as follows:

a) First, the training step that provides the system with known scores essays.

b) Second, the validation step examines the scoring model against a smaller set of known scores essays.

c) Finally, application to new essays with unknown scores.

IntelliMetric identifies text related characteristics as larger categories called Latent Semantic Dimensions (LSD). Figure 2.2 represents the IntelliMetric features model.

IntelliMetric scores essays in several languages including English, French, German, Arabic, Hebrew, Portuguese, Spanish, Dutch, Italian, and Japanese . According to Rudner, Garcia, and Welch , the average of the correlations between IntelliMetric and human-raters was 0.83 .

Fig 2.2 The IntelliMetric features model.

2.2.5 Criterion

Criterion is a web-based scoring and feedback system based on ETS text analysis tools: E-rater R and Critique. As a text analysis tool, Critique integrates a collection of modules that detect faults in usage, grammar, and mechanics, and recognizes discourse and undesirable style elements in writing. It provides immediate holistic scores as well. Criterion similarly gives personalized diagnostic feedback reports based on the types of assessment instructors give when they comment on students' writings. This component of the Criterion is called an advisory component. It is added to the score, but it does not control it. The types of feedback the advisory component may provide are like the following:

- The text is too brief (a student may write more).
- The essay text does not look like other essays on the topic (the essay is off-topic).
- The essay text is overly repetitive (student may use more synonyms).

2.3 PROPOSED SYSTEM

2.3.1 Deep Learning

Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for a couple of years now. It's on hype nowadays because earlier we did not have that much processing power and a lot of data. As in the last 20 years, the processing power increases exponentially, deep learning and machine learning came in the picture.

A formal definition of deep learning is-

Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones.

In human brain approximately 100 billion neurons all together this is a picture of an individual neuron and each neuron is connected through thousand of their neighbours. The question here is how do we recreate these neurons in a computer. So, we create an artificial structure called an artificial neural net where we have nodes or neurons. We have some neurons for input value and some for output value and in between, there may be lots of neurons interconnected in the hidden layer.

Deep learning is the subfield of machine learning, supporting algorithms that are inspired by the structure and function of the human brain, and named as artificial neural networks. Deep learning is the one category of machine learning that emphasizes training the computer about the basic instincts of human beings.

In deep learning, a computer algorithm learns to perform classification tasks directly on complex data in the form of images, text, or sound. These algorithms can accomplish state-of-the-art (SOTA) accuracy, and even sometimes surpassing human-level performance. They are trained with the large set of labeled data and neural network architectures, involving many layers. Moreover;

1. Deep Learning is a prime technology behind the technology such as virtual assistants, facial recognition, driverless cars, etc.
2. The working of deep learning involves training the data and learning from the experiences.
3. The learning procedure is called ‘Deep’, as with every passing minute the neural networks rapidly discover the new levels of data. Each time data is trained, it focuses on enhancing the performance.
4. With the increasing depth of the data, this training performance and deep learning capabilities have been improved drastically, and this is because it is broadly adopted by data experts.

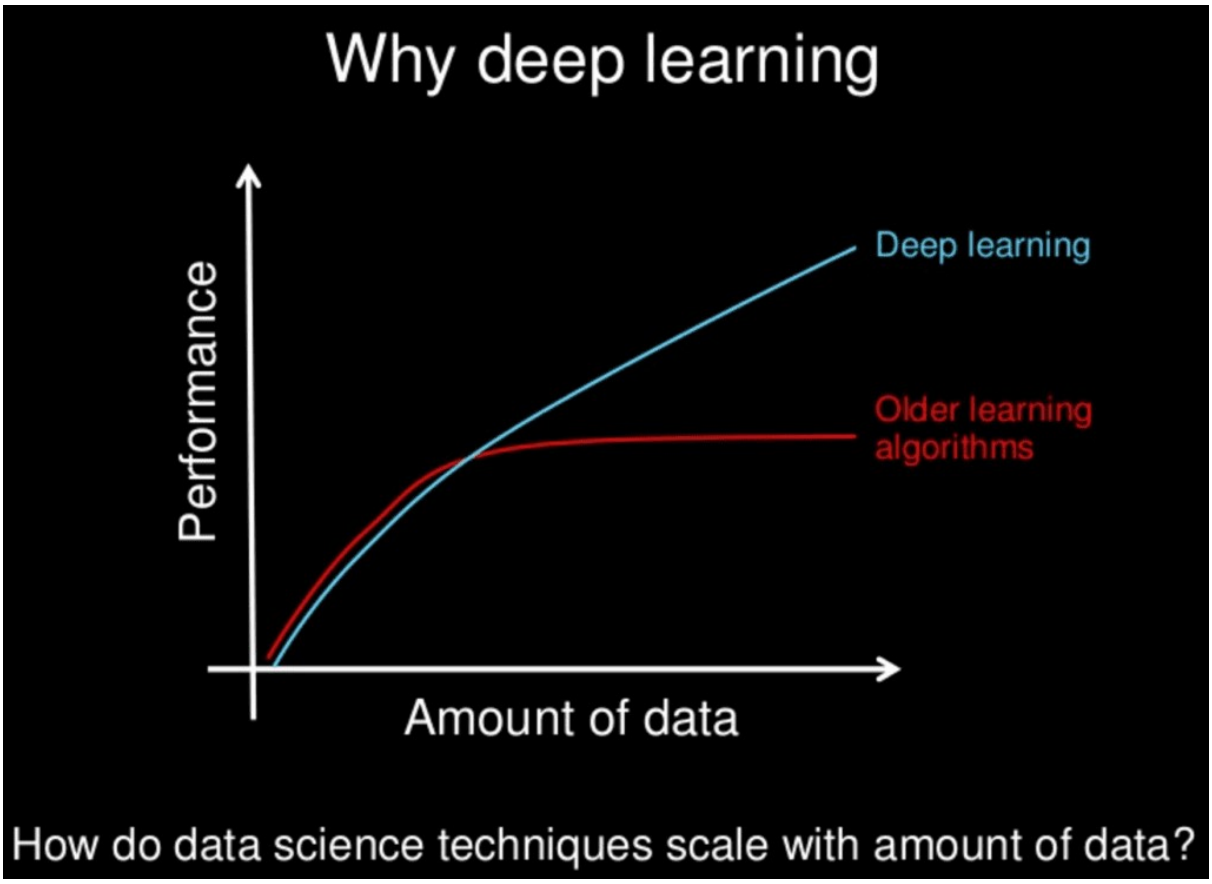


Fig 2.3 Why to choose Deep Learning.

Along with the ample amount of benefits, threats also surfaces due to the unexplored capabilities of deep learning.

Deep learning utilizes supervised, semi-supervised and unsupervised learning to train from the data representations. The functionality of deep learning relies on the below points:

- It imitates the functionality of a human brain for managing the data and forming the patterns for referring it in decision making.
- The trained dataset can be interconnected, diverse and complex in nature.

- The larger the data set, the more efficient the training that directly impacts the decision making.

Advantages of Deep Learning

- Ability to generate new features from the limited available training data sets.
- Can work on unsupervised learning techniques helps in generating actionable and reliable task outcomes.
- It reduces the time required for feature engineering, one of the tasks that requires major time in practicing machine learning.
- With continuous training, its architecture has become adaptive to change and is able to work on diverse problems.

Disadvantages of Deep Learning

With the increasing popularity, deep learning also has a handful of threats that needs to be addressed:

- The complete training process relies on the continuous flow of the data, which decreases the scope for improvement in the training process.
- The cost of computational training significantly increases with an increase in the number of datasets.
- Lack of transparency in fault revision. No intermediate steps to provide the arguments for a certain fault. In order to resolve the issue, a complete algorithm gets revised.
- Need for expensive resources, high-speed processing units and powerful GPU's for training the data sets.

Working of Deep Learning

Deep learning algorithms utilizes supervised and unsupervised learning algorithms to train the outputs through the delivered inputs.

See the image below, these circles represent neurons that are interconnected. The neurons are classified into three different hierarchies of layers termed as Input, Hidden and Output Layers.

- The first neuron layer i.e. input layer receives the input data and passes it to the first hidden layer.
- The hidden layers perform the computations on the received data. The biggest challenge under neural networks creation is to decide the number of neurons and a number of hidden layers.

- Finally, the output layer produces the required output.

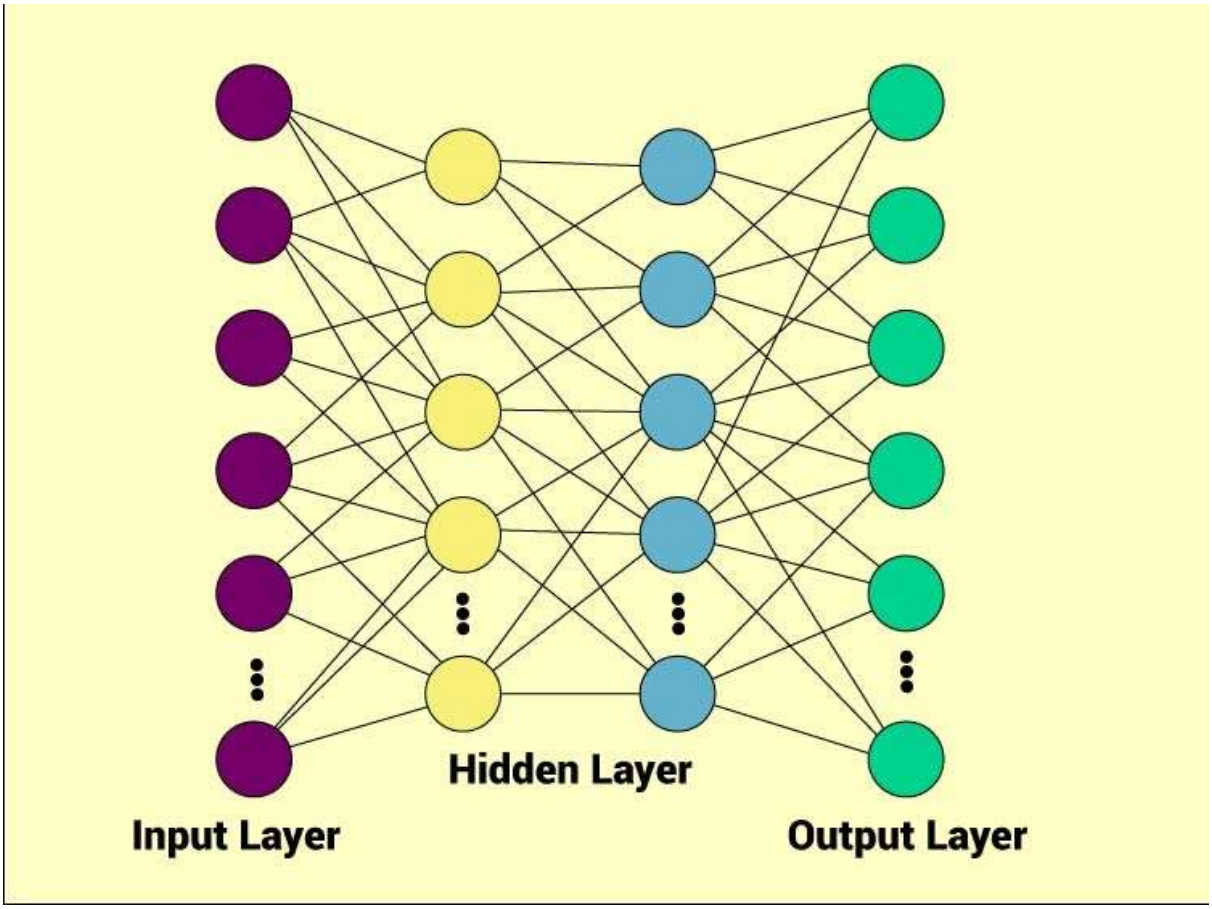


Fig 2.4 Working network of deep learning.

This is the basic flow of working. Now, comes the point where the method of computation is explained.

Every connection between the neurons consists of weights, it denotes the significance of the input values. In order to standardize the outputs, an activation function is used.

For training the network, two important measures are considered. The first is to create a large data set and the second is large computational power. The 'Deep' in deep learning signifies the number of hidden layers the model is using to train the data set.

Working of Deep learning can be summed up in four final points:

1. ANN asks a combination of binary True/False queries.
2. Extracting numeric values from blocks of data.
3. Sorting the data as per the received answers.
4. A final point is marking/labeling the data.

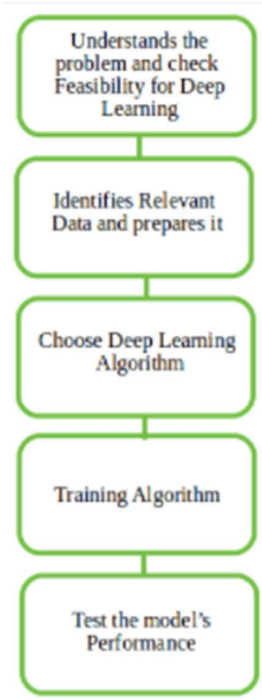


Fig 2.5 Working of Deep learning.

The greater the experience of deep-learning algorithms, the more effective they become. As the technology progresses over the years, it has the potential to become extraordinary.

CHAPTER-3
ANALYSIS

3.1 INTRODUCTION

The peculiarity of this problem is collecting the essay's real time working with the detection at the same time, so we developed an user interface who'll be accessing for the grade detection of their essay. First the text in the dataset is preprocessed by Natural Language Processing. Natural Language Processing (NLP) refers to AI method of communicating with an intelligent system using a natural language such as English. Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions, when you want to hear decision from a dialogue based clinical expert system, etc. The field of NLP involves making computers to perform useful tasks with the natural language's humans use.

Every day, we say thousand of a word that other people interpret to do countless things. We, consider it as a simple communication, but we all know that words run much deeper than that. There is always some context that we derive from what we say and how we say it., NLP in Artificial Intelligence never focuses on voice modulation; it does draw on contextual patterns.

Natural Language Generation (NLG) is the process of producing meaningful phrases and sentences in the form of natural language from some internal representation.

Text cleaning or pre-processing will be done using different libraries. "Re" is the library which is used to replace the selected special characters with desired parameter. "NLTK" – Natural language Tool Kit is the library used for stemming using a special class in the library. Punctuations, Numbers doesn't help much in processing the given text, so we will be using re library to replace all the punctuations numbers with a space while excluding alphabets. As in the dataset the reviews are present in Review column, we are declaring a variable called review and assigning the first row of the column to declared variable. Then using re library, we are substituting all the other special characters with a space excluding alphabets ([^a-zA-Z], this indicates except this replace everything with space).

Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers.10 To stem each word first we have to split the review in to a list and then apply stemming functionality. we use port stemmer class for stemming purpose. We are also removing stop words like "this", "that", "and", "is", "what" etc. "The syntax checks all the words in the list which is split, if the word is not a stop word, then you are applying stemming to stem the selected word" At first, we got like lot of wrong predictions accuracies because we tried more algorithms for best accurate algorithm. Finally, after we

implemented model using RNN and developed it to use as a real time application for the prediction of grades.

3.2 SOFTWARE REQUIREMENTS

- Jupyter Notebook Environment
- Spyder Ide
- Flask

We developed this automated essay grading by using the Python language which is a interpreted, dynamically typed and highlevel programming language and using the Machine Learning algorithms.

For coding we used the Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language.

For creating an user interface for the prediction we used the Flask. It is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a webpage in HTML by creating the templates to use in the functions of the Flask and HTML.

3.3 HARDWARE REQUIREMENTS

- Windows 7,8 or 10(32 or 64 bit)
- RAM-8GB
- Processor:1.5HZ or above
- HDD:100GB or above

3.3 ARCHITECTURE

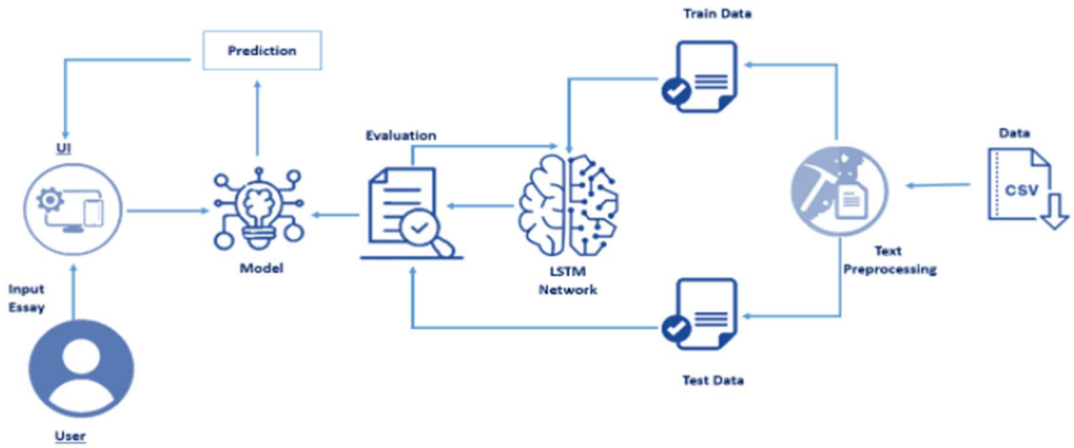


Fig 3.1 ARCHITECTURE

3.4 ALGORITHMS & FLOWCHART

The following are the deep learning techniques which we have used in this project.

3.4.1 NATURAL LANGUAGE PROCESSING

According to the focus of NLP tasks, the methods for NLP can be categorized into two types. This first one is Syntax Analysis. It relates to understand structures of word, sentence, and documents. Typical tasks include morphological segmentation, word segmentation, part of speed tagging (POS), and parsing. The second one is semantics analysis. It aims at understanding meanings of word, sentences and their combinations. Typical tasks include name entity recognition, sentiment analysis, machine translation, and question answering. There is another basic task that does not fall into the two categories above, which is the word/char imbedding. It aims to solve the problem of representing words/chars in vocabulary in vectors to enable applying machine learning techniques on the above two tasks. A typical method is the word2vec that projects words into vector spaces such that similarity of words can be measured using cosine similarity.

NLP aims at converting unstructured data into computer-readable language by following attributes of natural language. Machines employ complex algorithms to break down any text content to extract meaningful information from it. The collected data is then used to further teach machines the logics of natural language. Natural language processing uses syntactic and semantic analysis to guide machines by identifying and recognising data patterns.

There are the following steps to build an NLP pipeline -

Step1: Sentence Segmentation

Sentence Segment is the first step for building the NLP pipeline. It breaks the paragraph into separate sentences.

Step2: Word Tokenization

Word Tokenizer is used to break the sentence into separate words or tokens.

Step3: Stemming

Stemming is used to normalize words into its base form or root form. For example, celebrates, celebrated and celebrating, all these words are originated with a single root word "celebrate." The big

problem with stemming is that sometimes it produces the root word which may not have any meaning.

Step 4: Lemmatization

Lemmatization is quite similar to the Stamming. It is used to group different inflected forms of the

word, called Lemma. The main difference between Stemming and lemmatization is that it produces the root word, which has a meaning.

Step 5: Identifying Stop Words

In English, there are a lot of words that appear very frequently like "is", "and", "the", and "a". NLP pipelines will flag these words as stop words. **Stop words** might be filtered out before doing any statistical analysis.

Step 6: Dependency Parsing

Dependency Parsing is used to find that how all the words in the sentence are related to each other.

Step 7: POS tags

POS stands for parts of speech, which includes Noun, verb, adverb, and Adjective. It indicates that how a word functions with its meaning as well as grammatically within the sentences. A word has one or more parts of speech based on the context in which it is used.

Step 8: Named Entity Recognition (NER)

Named Entity Recognition (NER) is the process of detecting the named entity such as person name, movie name, organization name, or location.

Step 9: Chunking

Chunking is used to collect the individual piece of information and grouping them into bigger pieces of sentences.

3.4.2 RNN-LSTM (LONG - SHORT TERM MEMORY)

Since the order matters a lot in understanding a chuck of text, it is natural to design a network in order to ‘remember’ orders of each input word and keep key information in memory. Recurrent neural network is a network designed to capture temporal behavior. As shown in Figure , a chunk of

RNN loops inside itself. With each coming input, it updates network parameters considering current input (Xt) and previous state (ht-1). As length of input gets longer, however, it is more time-consuming to train a RNN. Moreover, more recent information may not be more important necessarily. It requires RNN to keep more “long-term” information in memory.

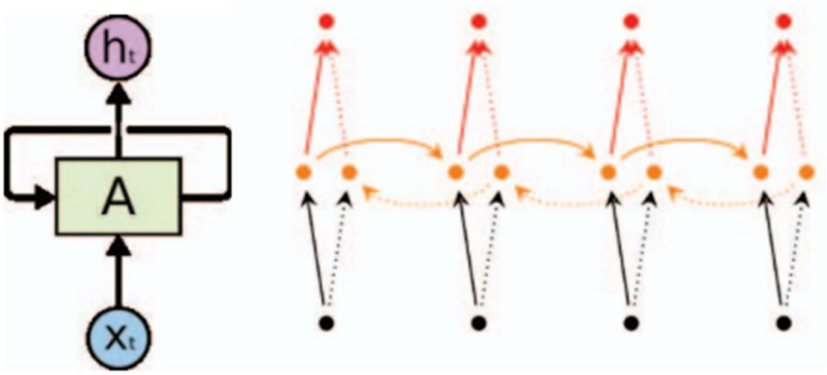


Fig 3.2 RNN chunk, connected RNN chunks, stacks RNN chunks, bidirectional RNN

To solve this problem, researchers introduced a new type of RNN - Long-short term memory (LSTM). The main merit is the introduction of a structure called gates. LSTMs can use gates to decide whether to let information through and drop certain information when necessary. One famous variation of LSTM called Gated Recurrent Unit (GRU) is firstly used in Cho et al. Due to the success of GRU, more types of gates are introduced such as input gate, forget gate, and so on.

Note that RNN chucks can be connected to others or stacked to form deep network. The direction of forwarding data can be forward, backward or bidirectional. Greff et al. compared 8 LSTM-based models on three tasks: speech recognition, handwriting recognition, and polyphonic music modeling. Jozefowicz et al. targeted at finding out whether RNN architecture is optimized. They experimented on ten thousand’s of RNN architectures and found one RNN architecture that outperformed both LSTM and GRU. Also, another notable variant is recursive neural network.

LSTM stands for long short-term memory networks, used in the field of Deep Learning. It is a variety of recurrent neural networks (RNNs) that are capable of learning long-term dependencies, especially in sequence prediction problems. LSTM has feedback connections, i.e., it is capable of processing the entire sequence of data, apart from single data points such as images. This finds application in speech recognition, machine translation, etc. LSTM is a special kind of RNN, which shows outstanding performance on a large variety of problems.

LSTM networks are indeed an improvement over RNNs as they can achieve whatever RNNs might achieve with much better finesse. As intimidating as it can be, LSTMs do provide better results and are truly a big step in Deep Learning. With more such technologies coming up, you can expect to get more accurate predictions and have a better understanding of what choices to make.

3.4.3 FLOWCHART

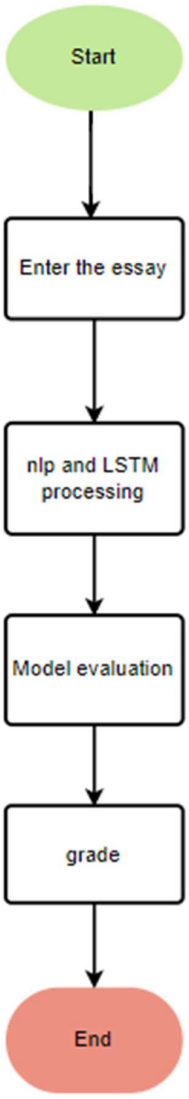


Fig 3.3 Flow chart

CHAPTER-4

DESIGN

An in-depth explanation of the experiments conducted on the data set is given in this section.

4.1 DATA PREPARATION

Hewlett has sponsored automated student assessment prize competition to get fast, effective and affordable solutions for automated grading of the student-written essay. They have provided access to hand scored essays. For this competition, there are eight essay sets. Selected essays range from an average length of 150 to 550 words per response. All responses were written by students ranging in grade levels from Grade 7 to Grade 10. All essays were hand graded and were double scored.

The Hewlett essay scoring dataset (The Hewlett Foundation: Automated Essay Scoring) was used in this research. The dataset has eight essay sets, which are handwritten by students ranging in grade levels from Grade 7 to Grade 10. All essays were hand graded and were double scored. The training data contained essay sets 1-8. Each essay set had a description and a rubric for the score, in this project all 8-essay set were used. The description of each set is as follows:

Essay set #1 was written by grade level 8 students. The type of essays is persuasive/narrative/expository, the training set size contains 1785 essays, the average length of essays are 350 words. The set is evaluated by two raters, rater1 and rater2 who gave score1 and score2. The rubric range for essay set #1 is 1-6. This set consists of a resolved score which is the sum of the scores of both raters which ranges from 2-12.

Essay set #2 was written by students of grade-level 10, the type of essays is persuasive/narrative/expository, the training set size contains 1800 essays, the average length of essays are 350 words. The set was evaluated by two raters from two domains; domain1 is evaluated on writing application, i.e. the rubric is based on the ideas and content, the organization, style and voice of the essay which is evaluated on the rubric 14 range 1-6 by rater1 and rater2 of domain1. Domain2 was evaluated on the language conventions of the written essay, the rubric range is 1-4 graded by both rater1 and rater2 of domain2. In this project, domain1 scores are taken into consideration.

Essay set #3 was written by students from grade level 10, the type of essays is source dependent responses, the size of the training set is 1726 essays, the average length of essays is 150

words. The set is evaluated by two raters; rater1 and rater2 based on a rubric range 0-3, the resolved score of both raters is the average of the rater1 and rater2 and is in the range is 0-3.

Essay set #4 consists of grade level 10 students' essays of type source dependent responses, the training set size is 1772 essays and the average length of essays are 150 words. The set is evaluated by rater1 and rater2 on the rubric range 0-3, has a resolved score which is the best score of the rater1 and rater2 and the resolved score range is 0-3.

Essay set #5 contains students' essays in grade level 8, the type of essays are source dependent responses, the set consists of 1805 training essays with an average essay length of 150 words. The essay set #5 is graded by both the raters, rater1 and rater2. The score range of the rater1 and rater2 is 0-4, has a final score, which is the average of the rater1 and rater2, final has a rubric range 0-4.

Essay set #6 was written by students of grade level 10, the type of essays is source dependent responses, the training set size is 1800 essays and the average length of essays is 150 words. The set is evaluated by rater1 and rater2 based on the rubric range 0-4, which has a final score equals to the average of rater1 and rater2 with final score range 0-4.15.

Essay set #7 was written by grade level 7 students. The type of essays is persuasive/narrative/expository, the training set size contains 1730 essays, the average length of essays is 250 words. The set is evaluated based on different parameters like ideas, organization, style and conventions. Evaluation was done by two raters; rater1 and rater2 who gave score1 and score2. The rubric range for essay set #7 is 0-15. This set consists of a resolved score which is the sum of the scores of both raters with the range 0- 30.

Essay set #8 was written by grade level 10 students the type of essays is persuasive/narrative/expository, the training set size contains 918 essays, the average length of essays is 650 words. The set is evaluated on different parameters like the ideas and content, organization, voice, word choice, sentence Fluency, style and conventions by three raters; rater1, rater2 and rater3 who gave score1, score2 and score3. The rubric range for essay set #8 is 0-30. This set consists of a resolved score which is composite of the scores of three raters and lies in the range 0-60.

In this project, the conducted experiments considered rater1 scores, rater2 scores and the average of rater1 and rater2 scores for all essay sets by preprocessing all the scores of the essay sets in the ranges 0-10.

4.2 UML DIAGRAMS

4.2.1 CLASS DIAGRAM

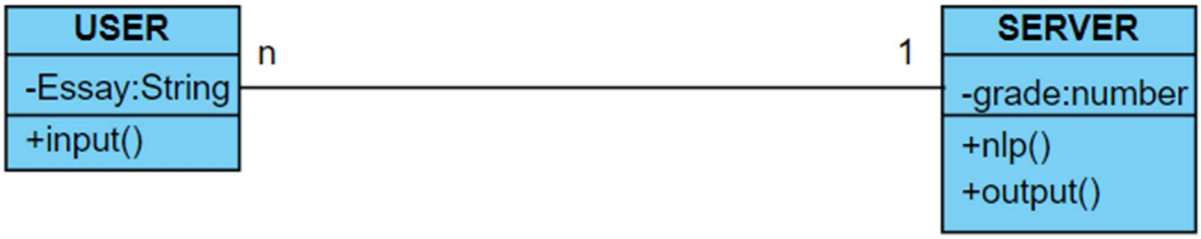


FIG 4.1: Class diagram

4.2.2 USE CASE DIAGRAM

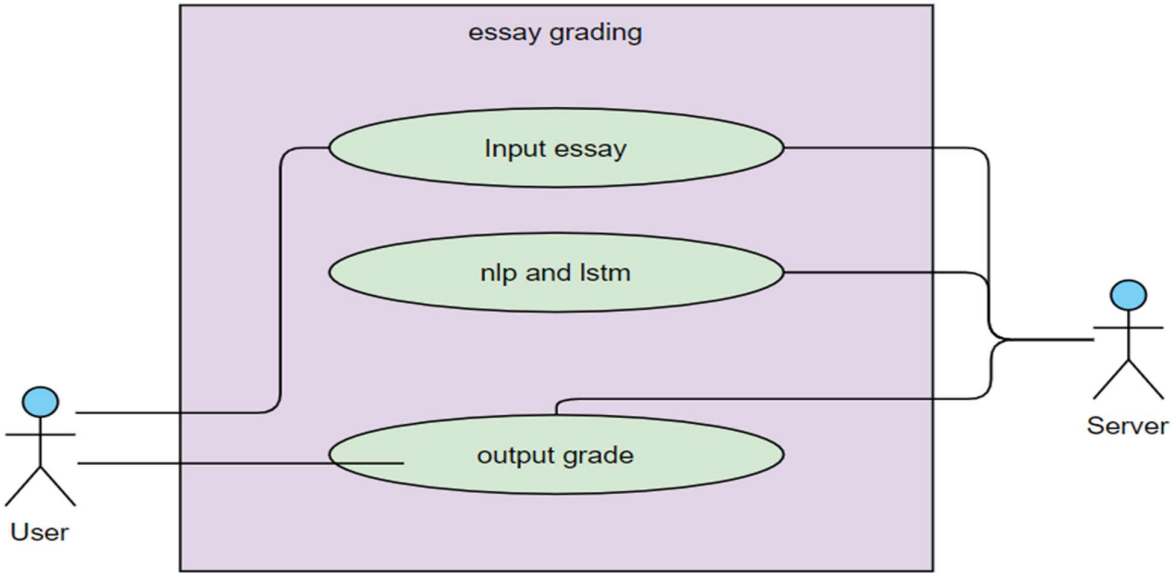


FIG 4.2: Use case diagram

4.2.3 ACTIVITY DIAGRAM WITH SWIMLANES

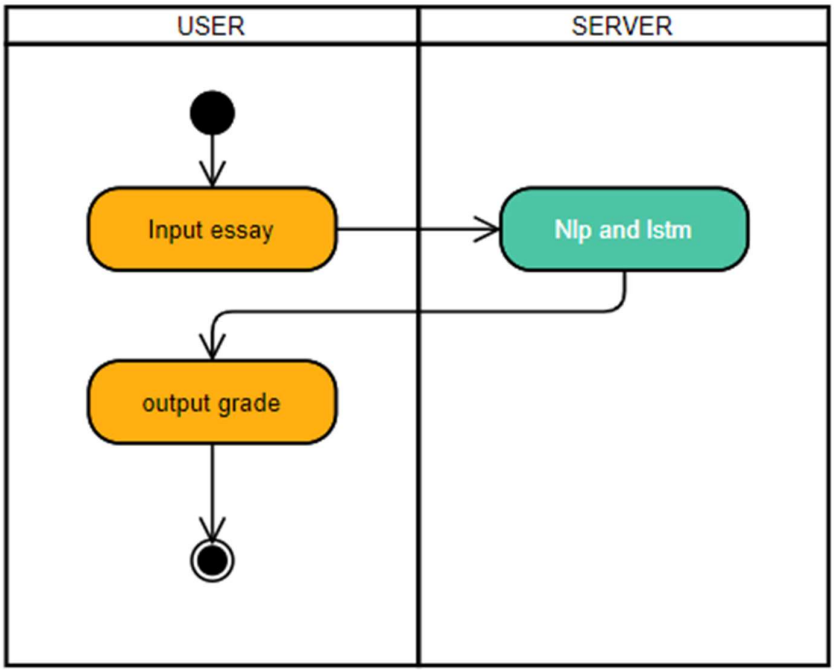


FIG 4.3: Activity diagram with swimlanes

4.2.4 SEQUENCE DIAGRAM

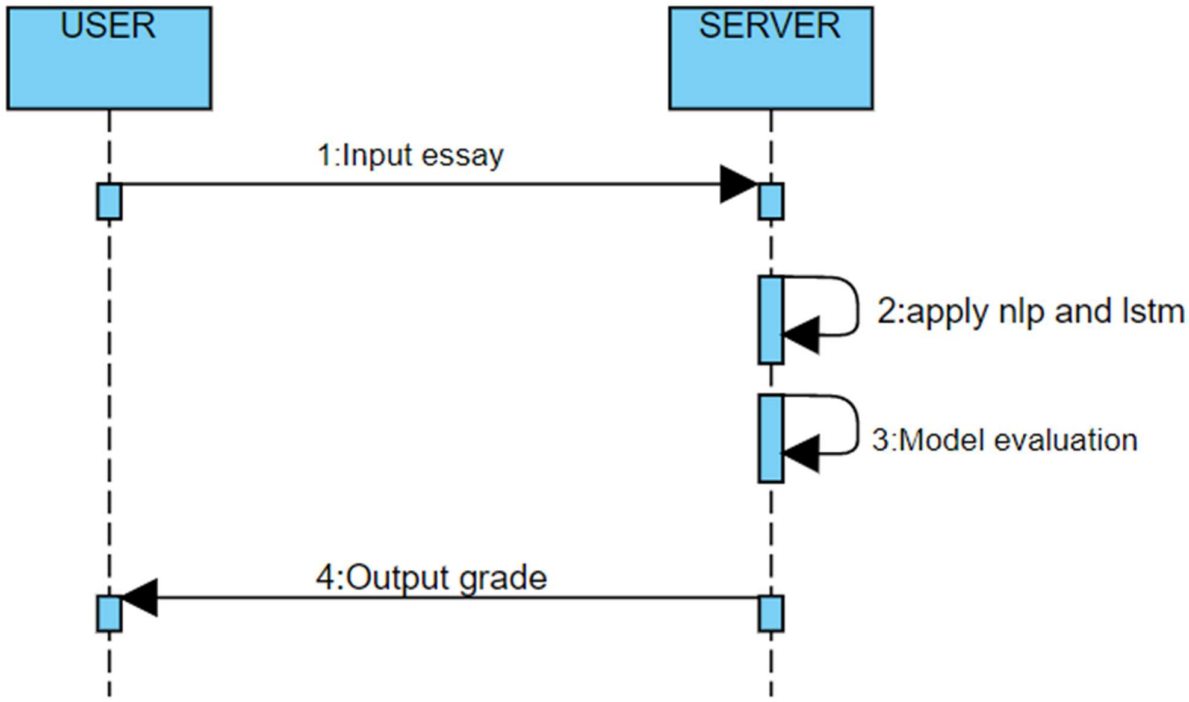


FIG 4.4: Sequence diagram

4.2.5 COMPONENT DIAGRAM

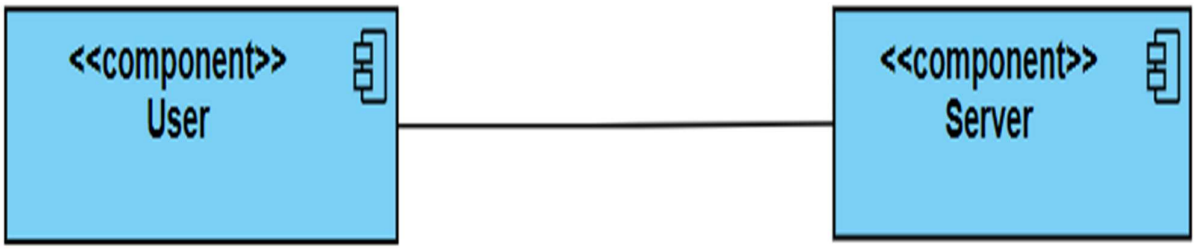


FIG 4.5: Component diagram

4.2.6 DEPLOYMENT DIAGRAM

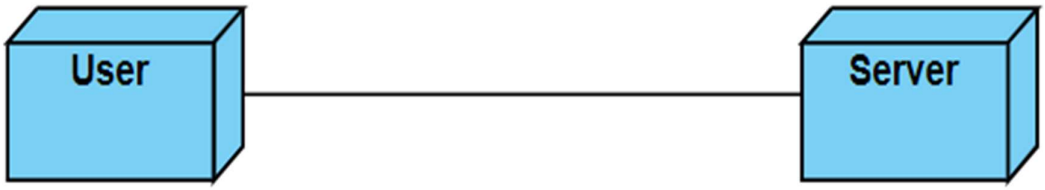


FIG 4.6: Deployment diagram

4.3 DESIGN OF PROJECT

Various deep learning techniques discussed in Chapter 3 are used as the basis for the experiments conducted in this project. They are described in detail below. We create a list of words from each sentence and from each essay. This list is fed into the Word2Vec model. This model makes sense of the available words by assigning numerical vector values to each word. Features are generated by passing the essays through Word2Vec model. The Word2Vec model acts as an Embedding Layer in a neural network. Features from this model are passed through our LSTM layers. We implement 2 LSTM layers. The first layer accepts all features from the Embedding Layer (Word2Vec) as input and passes 300 features as output to the second LSTM layer. The second layer accepts 300 features as input and 64 features as output. Next we add a Dropout layer with value 0.5. Finally a fully connected Dense Layer with output 1 which represents the score of Essay. The model was compiled with loss function Mean Squared Error and Optimizer Root Mean Square. The model was trained for

150 epochs with batch size of 64. They are described in detail below.

1.Training LSTM Model

Importing the Data

- Gensim, NLTK, libraries have been added.
- Constants have been added .
- Loading data using pandas library from training_set_rel3.tsv.
- Removing unnecessary columns like domain_score and raters_domain.
- Defining Minimum and Maximum score which we will be using at the time of predicting the actual score.

2.Preprocessing the Data

We will pre-process all essays and convert them to feature vectors so that they can be fed into the RNN.There are 4 functions defined:

1. **getAvgFeatureVecs**: This function accepts 3 parameters: essays, model, num_features. It internall calls **makeFeatureVec** function to convert essays into FeatureVector.
2. **makeFeatureVec:** This function accepts 3 parameters: words, model, num_features. Using Word2Vec index2word function and np.divide it gives ultimately average feature vectors for the passed model.
3. **essay_to_sentense**: This function accepts 2 parameters: essay_v, remove_stopwords. it internally calls essay_to_wordlist and converts essays to sentences.
4. **essay_to_wordlist**: This function accepts 2 parameters: essay_v, remove_stopwords. It removes the stopwords and returns words.

- Whenever you are working with NLP Machine Learning and Deep Learning tasks the above mentioned steps are almost neccessary because machine understands numbers or we can say that computation is very easy when we use numbers here we refer to vectors.
- We are trying to convert essay or corpus to first sentenses and then to words which can also be called are tokens and then convert them to vectors.

3.Defining the model

Here we define a 2-Layer LSTM Model.

Instead of using sigmoid activation in the output layer we will use Relu since we are not normalising training labels.

4.Training Phase

Now we train the model on the dataset.

We will use 5-Fold Cross Validation and measure the accuracy.

- We are first training the essays using Word2Vec model which is available in gensim library. Later on, we are saving into word2vecmodel.bin file which we will be using at the time of predicting the score.
- Now we are using the fuctions which we have previously defined to convert essay to vector representation.
- We are also passing this vectors into LSTM model and saving the model in final_lstm.h5 file.
- Atlast we have calculated accuracy.
- And built the UI .

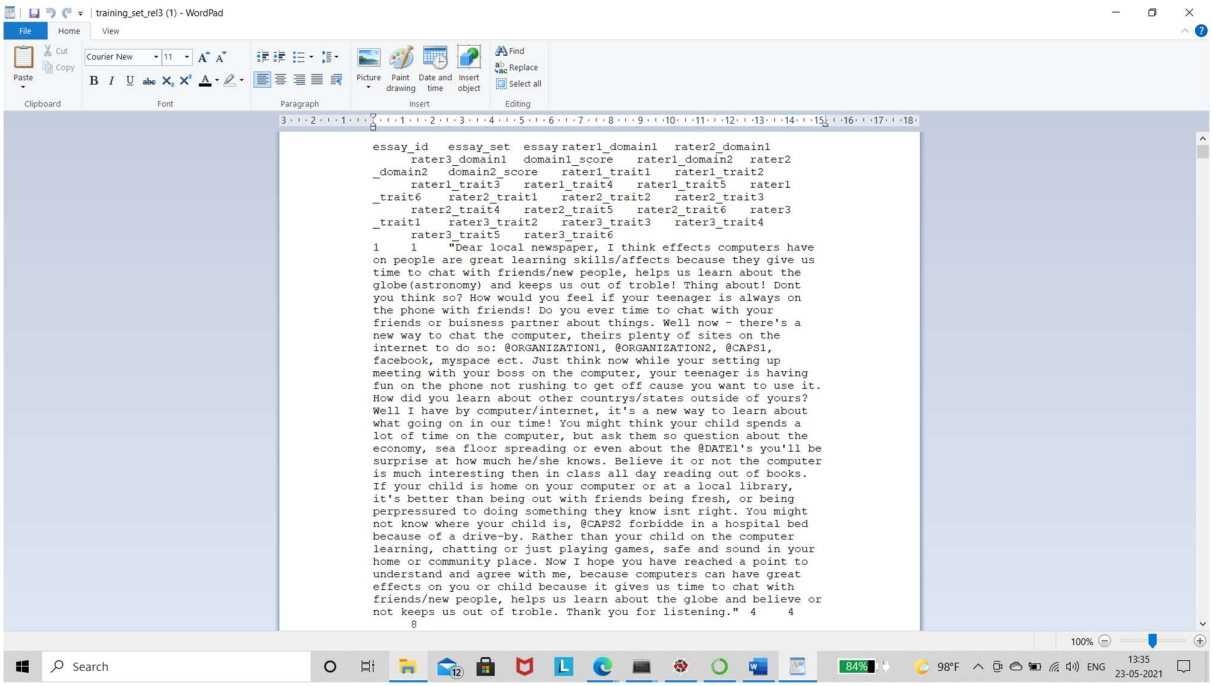
4.4 MODULE DESIGN AND ORGANIZATION

We used essays provided for an automated essay scoring competition sponsored by the Hewlett Foundation. The data were divided into eight essay sets. The authors of the essays were American students in grades seven through ten. The essay sets had an average essay length between 150 and 650 words. Each dataset used a different prompt; some of the prompts asked for responses to source material while the rest asked students to respond to a short statement. Each essay was graded by at least two humans. Each essay set had a procedure for producing a final score if the two human scores disagreed, e.g. take the average, or use a third human score as a mediator.

The training data is provided in three formats: a tab-separated value (TSV) file, a Microsoft Excel 2010 spreadsheet, and a Microsoft Excel 2003 spreadsheet. The current release of the training data contains essay sets 1–6. Sets 7–8 will be released on February 10, 2012.

- essay_id: A unique identifier for each individual student essay

- essay_set: 1–8, an id for each set of essays
- essay: The ascii text of a student's response
- rater1_domain1: Rater 1's domain 1 score; all essays have this
- rater2_domain1: Rater 2's domain 1 score; all essays have this
- rater3_domain1: Rater 3's domain 1 score; only some essays in set 8 have this.
- domain1_score: Resolved score between the raters; all essays have this
- rater1_domain2: Rater 1's domain 2 score; only essays in set 2 have this
- rater2_domain2: Rater 2's domain 2 score; only essays in set 2 have this
- domain2_score: Resolved score between the raters; only essays in set 2 have this rater1_trait1 score — rater3_trait6 score: trait scores for sets 7–8



Screen 4.1 Dataset

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 INTRODUCTION

Project implementation (or project execution) is the phase where visions and plans become reality. This is the logical conclusion, after evaluating, deciding, visioning, planning, and finding the resources of a project. Technical implementation is one part of executing a project. Results basically refer to any particular output or end point that comes as a result of the completion of the activities and or processes that have been performed as part of the project or as part of a particular project component.

This chapter encompasses data pre-processing methods and implementation steps of this work.

5.2 METHOD OF IMPLEMENTATION

Our approach to tackle this problem involves the following steps:

5.2.1. Data Gathering

We have acquired our data from the the William and Flora Hewlett Foundation from the Kaggle.com (as mentioned in the above section: Dataset). We have collected a total of around 12000 essays in which each essay is in the ASCII text format. The approximate length of each essay is around 150 to 550 words and is ideal for this project. We made use of only 8000 essays form the given dataset.

5.2.2 Data pre-processing

To pre-process the data, we have imported and used a package belonging to Python programming language called NLTK (Natural Language Toolkit). In data pre-processing, we first remove all numbers, whitespaces and default stop words (will, being, so, few, as, yours, had, have, and, not). Stop words are words which do not play a part in the meaning of a sentence. So, it makes sense to remove them as they do not possess much value to the meaning. Then we split the “cleaned essay” into tokens.

From this, we extract features like word count, character count, average word length, misspelled words, prevalence of the submitted essay and POS tagging. To get the misspelled word count, we have compared our data with a text file called big.txt which consists of large collection of words. As deep learning or machine learning models cannot understand text data when given as input, we have to convert out text into a format which the model can understand and take in to process it,

which is a numerical format or vector format. To produce feature vectors, we have used model architecture from word2vec called Continuous Bag Of Words (CBOW) which takes in text corpus as input and pops out feature vectors as its output. The cause and usefulness of Word2vec is to group the vectors of comparable words together in vector space. That is, it detects similarities mathematically. Word2vec model creates vectors which might be allotted numerical representations of word functions, features consisting of the context of character words.

5.2.3. Training Model

To train the model we, 5 fold cross validation is applied on the dataset. The model used here is a deep learning model called Sequential model. The reason to choose Sequential model is that it is a simple model which is just a linear arrangement of layers chosen. We can add our layers in the order we want to perform our computations. The layers we have implemented are 2 LSTM (Long short-term memory) layers and a single dense layer.

LSTM stands for Long-Short Term memory layer which is artificial recurrent neural network architecture. By stacking or using a 2 layered LSTM model, we have multiple hidden memory cells. So our networks become deeper thus allowing our network to perform better as the success of the learning sometimes depends on the depth.

A dense layer is a simple regular layer of neurons in a neural network. Each neuron takes the input from all the neurons in the previous layer, thus fully connected. We have also used Dropout technique with a value of 0.5 thus enabling it to drop a fraction of neurons to minimize overfitting as much as possible.

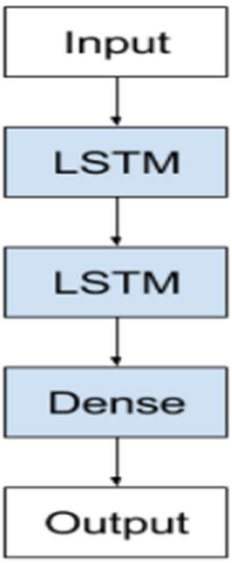


Fig 5.1 Model Architecture.

In the output layer we have used relu activation function (Rectified Linear Unit or ramp function) as

no normalizing of training labels is required. The output layer then produces the output of the essay as a discrete value or a single integer.

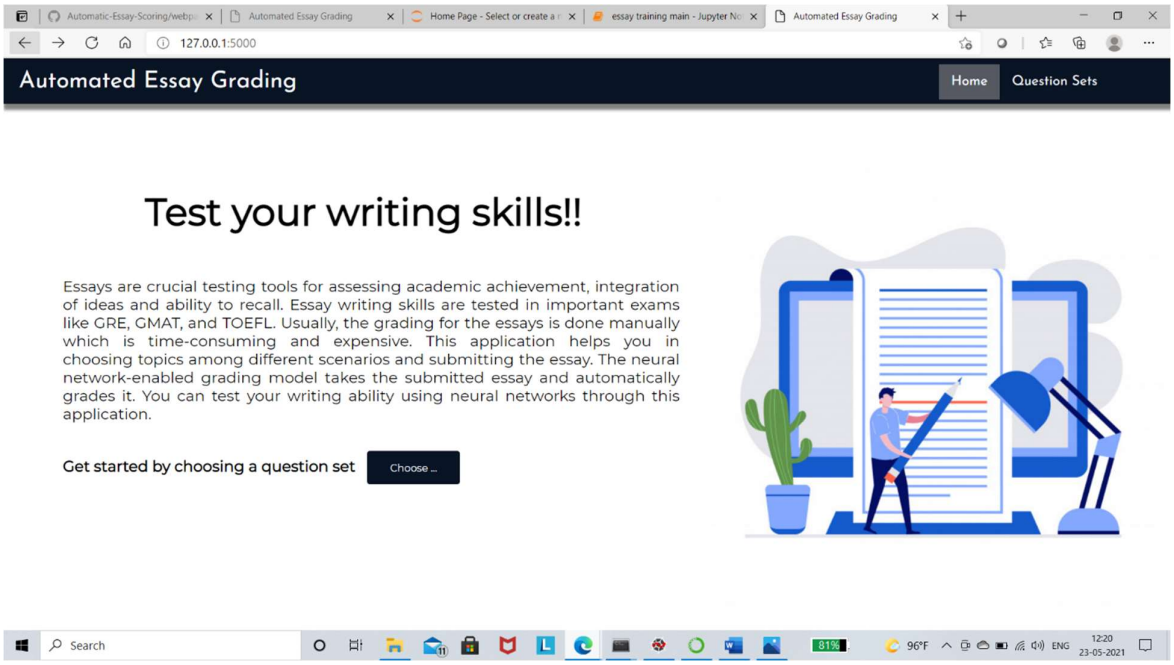
5.3 INPUT & OUTPUT SCREENS

Flask is a web application framework written in python, it makes the process of designing a web application simpler. Flask lets us focus on what the users are requesting and what sort of response to give back. Flask depends on the Jinja template engine and the Werkzeug WSGI toolkit.

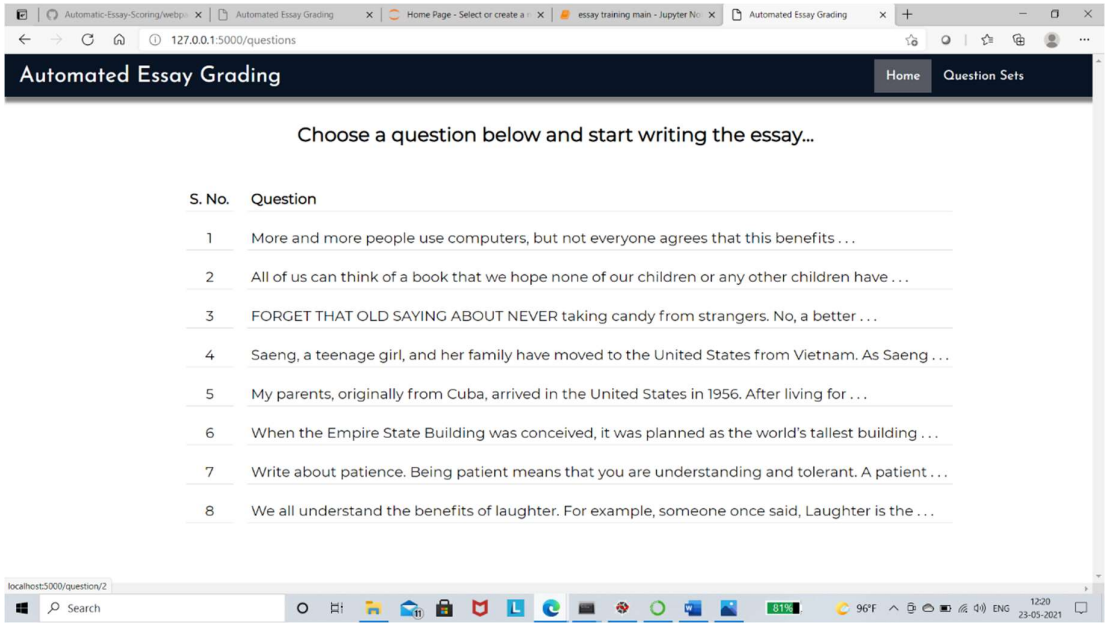
Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

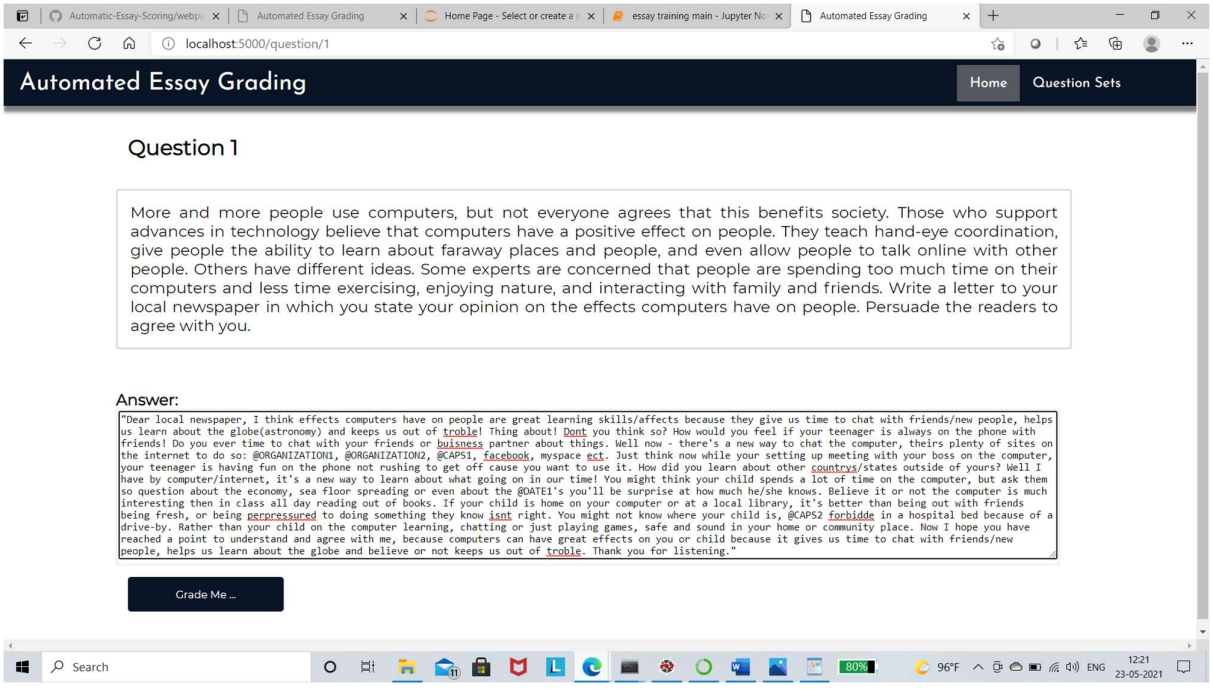
we have created an UI using the flask for grading the essay. The UI which we have created is shown below. When we save and run, the webpage will be opened in the local host. When the end user enter essay and submit, the grade is given. Our User interface displays all the attributes which are required for assigning the grade. when the end user gives the essay then based on the training data the UI will display the grade scored by the end user.



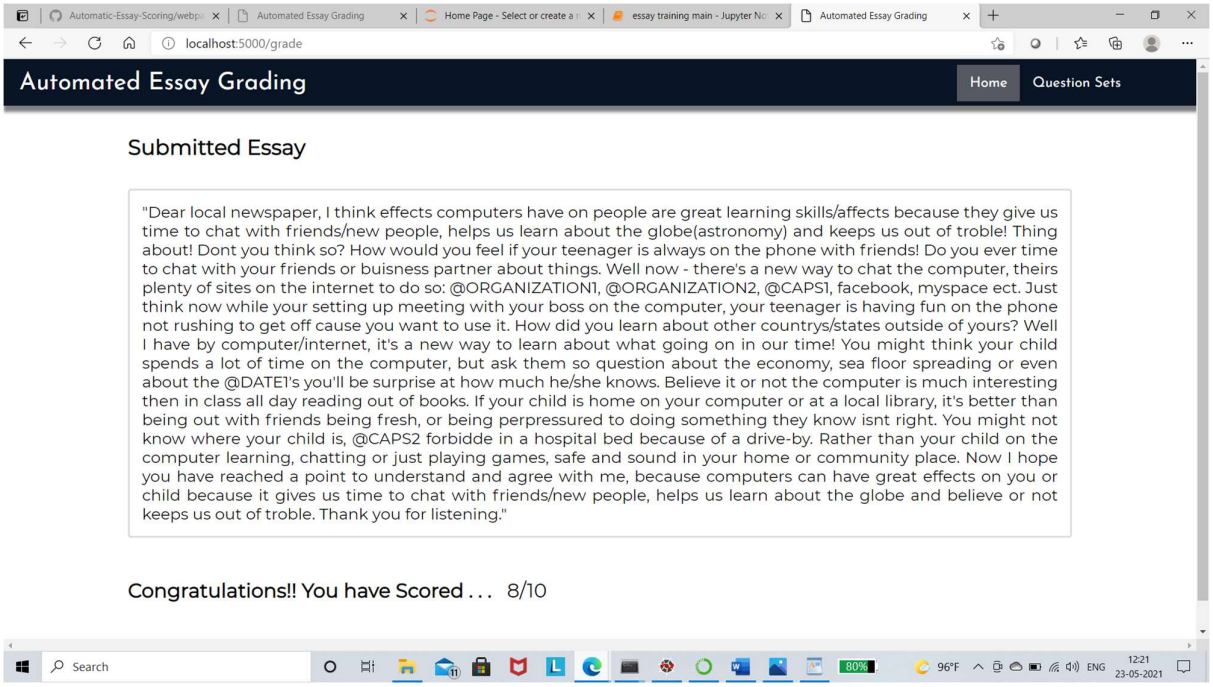
Screen 5.1 Input 1



Screen 5.2 Input 2



Screen 5.3 Input 3



Screen 5.4 Output

5.4 RESULT ANALYSIS

The essay taken as input will be pre-processed i.e. numbers, symbols and stopwords are removed; the cleaned essay will be converted into feature vectors. The feature vectors will be passed as input to the neural network consisting of the layers and the score or grade of the essay according to the features considered (word count, character count, average word length, misspelled words, and prevalence) will be displayed on the screen

In our project, assigning the grades for essay's using deep learning techniques is adopted to build a UI model for assigning grades based on the training data. The model shows that Recurrent Neural Networks-LSTM performs best in predicting the grades. There is no definitive guide of which algorithms to be used. What may work on some datasets may not work on others. Therefore, always check the accuracy and predict with the dataset values. The deep learning model is used to predict its performance, and compared with the results in the dataset, it gave accurate and reliable outputs.

CHAPTER 6

TESTING & VALIDATION

6.1 Introduction:

In order to test RNN model, tester defines three different datasets viz. Training dataset, validation dataset and a test dataset (a subset of training dataset). The models will be validated on validation and test sets of data for efficient detections. Seven metrics will be used for model evaluation, including accuracy. This study could potentially grade the essay's.

6.2 Design of test cases and scenarios:

Here, below is the basic approach a tester can follow in order to test the developed learning algorithm:

- Tester first defines three datasets, training dataset (65%), validation dataset (20%) and test dataset (15%). Then randomize the dataset before splitting and don't use the validation/test dataset in your training dataset.

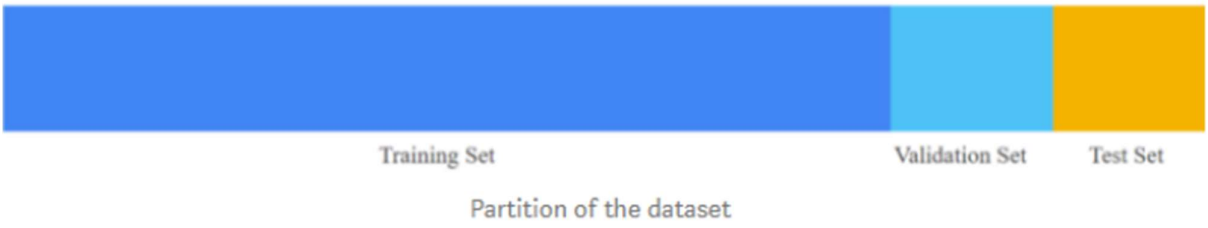


Fig 6.1 Partition of the dataset.

- Tester once defines the data set, Will begin to train the models with the training dataset. Once this training model is done, the tester then performs to evaluate the models with the validation dataset. This is iterative and can embrace any tweaks/changes needed for a model based on results that can be done and re-evaluated. This ensures that the test dataset remains unused and can be used to test an evaluated model.
- Once the evaluation of all the models is done, the best model that the team feels confident about based on the least error rate and high approximate prediction will be picked and tested with a test dataset to ensure the model still performs well and matches with validation dataset results. If you find the model accuracy is high then you must ensure that test/validation sets are not leaked into your training dataset.

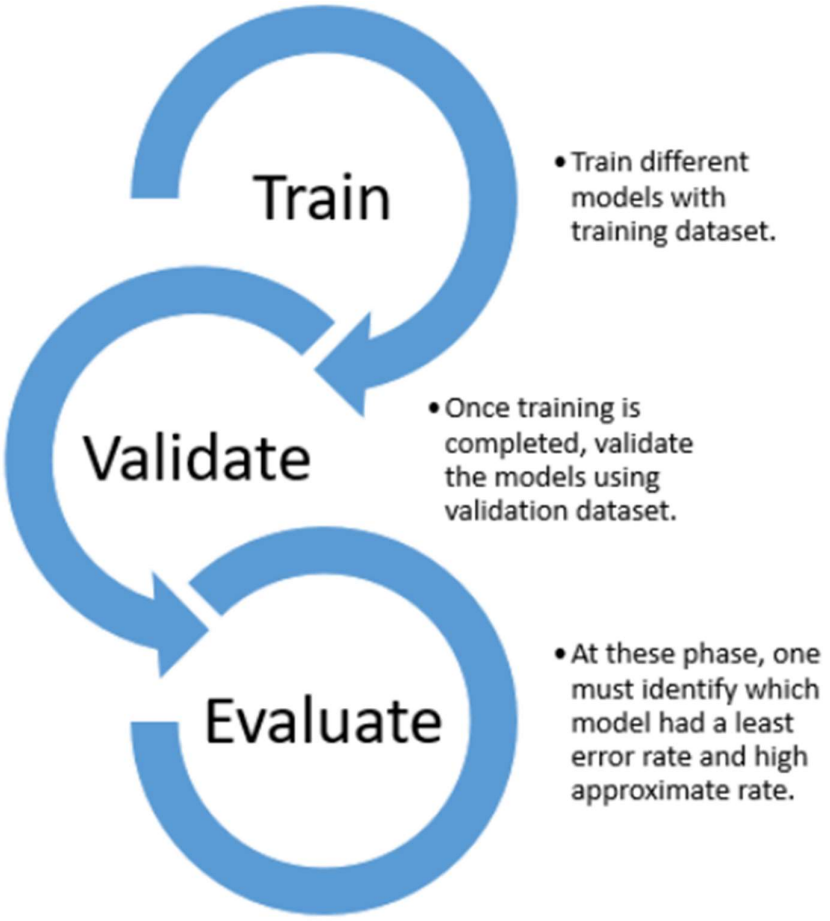


Fig 6.2 Process of training and testing.

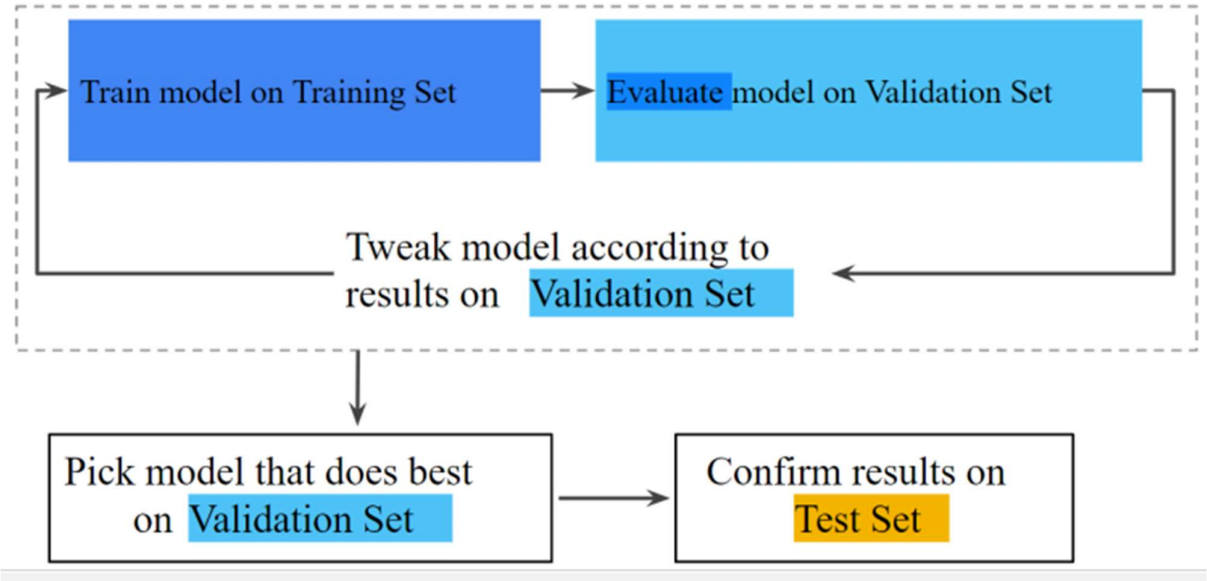


Fig 6.3 Process of selecting a model for the project.

6.3 Validation:

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs.

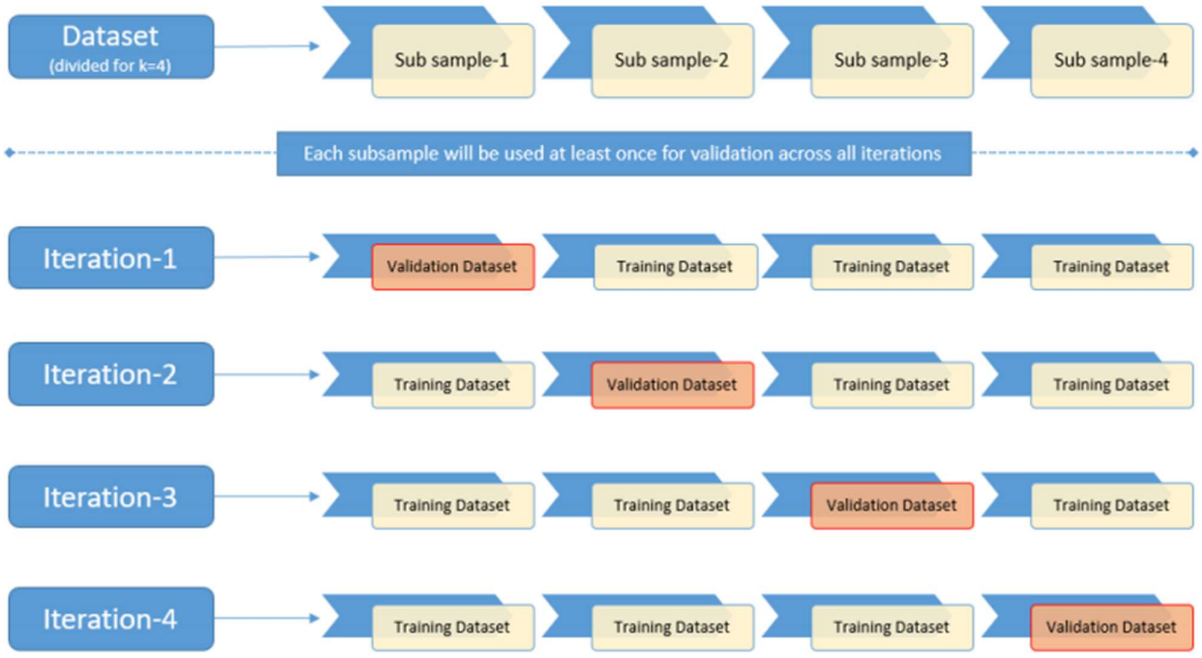


Fig 6.4 Dividing the dataset into K-subsets.

In this, the dataset is divided into k-subsets(folds) and are used for training and validation purpose for k iteration times. Each subsample will be used at least once as a validation dataset and the remaining (k-1) as the training dataset. Once all the iterations are completed, one can calculate the average prediction rate for each model.

CHAPTER 7

CONCLUSION

7.1 PROJECT CONCLUSION

The essay grading task is a laborious task which can be automated with the help of Deep Learning. The grader which is proposed in the paper has an accuracy of 74% which is the highest that has been observed until now. The grader is very good in differentiating between ambiguous sentence formations and the new vector representation is very good in implementing the NLP translations.

7.2 FUTURE ENHANCEMENT

The essay grader can be used by teachers for grading student essays. It can also be used by testing agencies for test of English writing to take the burden off the human graders. The concept can be extended to other languages too if the dataset is available. Lastly, the accuracy of the model can still be increased if more data is feeded to the LSTM network while training. More sophisticated models can be developed using transfer learning at the cost of using more resources required to train the model.

REFERENCES:

1. Adamson Alex, Andrew Lamb, and Ralph Ma, Automated Essay Grading. 2014
2. Higgins Derrick, Jill Burstein, Daniel Marcu, and Claudia Gentile, Evaluating Multiple Aspects of Coherence in Student Essays. In HLT-NAACL, pp. 185-192. 2004.
3. Gang Kou, Yi Peng, An Application of Latent Semantic Analysis for Text Categorization In International Journal of Computers, Communications & Control (IJCCC) 10(3):357 April 2015
4. Kaveh Taghipour, Hwee Tou Ng, A Neural Approach to Automated Essay Scoring In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, November 2016
5. Isaac Persing and Vincent Ng , Modeling Argument Strength in Student Essays , 2015.
6. Dimitrios Alikaniotis , Helen Yannakoudakis and Marek Rei , Automatic Text Scoring Using Neural Networks , 16 Jun 2016.
7. Ronan Collobert and Jason Weston NJ 08540, A Unified Architecture for Natural Language Processing,2011.
8. Peter Phandi1 , Kian Ming A. Chai2 and Hwee Tou Ng1 , Flexible Domain Adaptation for Automated Essay Scoring Using Correlated Linear Regression , 17-21 September 2015.
9. Kaggle. ”Develop an automated scoring algorithm for student-written essays.” (2012). https://www.kaggle.com/c/asap-aes
10. Drolia, S., et al., Automated Essay Rater using Natural Language Processing. International Journal of Computer Applications, 2017. 163(10)
11. Manvi Mahana, Mishel Johns, Ashwin Apte. Automated Essay Grading System using Machine Learning, CS229 Machine Learning-Autumn 2012.
12. Y.Harika, I.Sri Latha, V.Lohith Sai, P.Sai Krishna , M.Suneetha. Automated Essay Grading System using Feature Selection. p-ISSN: 2395-0072. Volume: 04 Issue: 03 | March -2017
13. Abhishek Suresh, Manuj Jha. Automated Essay Grading using Natural Language Processing and Support Vector Machine. IJCAT - International Journal of Computing and Technology, Volume 5, Issue 2, February 2018.
14. Peter W. Foltz, New Mexico State University, Darrell Laham, Knowledge Analysis Technologies.
15. S. Dikli, “Automated Essay Scoring”, Florida State University, Tallahassee (PEG).
16. Thomas K. Landauer, University of Colorado, “The Intelligent Essay Assessor: Applications to Educational Technology”, Volume 1, Number 2, October 1999.

17. V. Salvatore, F. Neri, and A. Cucchiarelli, "An overview of current research on automated essay grading.", Journal of Information Technology Education: Research 2.1 (2003): 319-330, 2003.
18. L. Hamp Lyons, “The Scope of writing assessment”, Elsevier Science Inc, 1075-2935/02 © 2002.
19. Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Landauer & Dumais, 1997; Landauer, Foltz, & Laham, 1998.
20. M. Rudner, V. Garcia, & C. Welch, “An Evaluation of the IntelliMetric™ Essay Scoring System Lawrence”, The Journal of Technology, Learning, and Assessment, Volume 4, Number 4 · March 2006.
21. https://en.wikipedia.org/wiki/Perplexity
22. https://www.kaggle.com/c/asap-aes

HELP FILE

The Theme of our Project is to detect the grades of essay using NLP and RNN-LSTM. A Artificial Neural Network model is developed to grade the essays and to make work easy for the teachers and universities and also saves the time for evaluation.

So, to predict this we need to use 4 codes i.e., Python code, Prediction code, Flask and HTML code. Here python code, prediction code needs to be executed in Jupyter notebook IDE python3 and Flask, HTML codes need to be executed in Spyder IDE. In python code, first we have to import the libraries and next preprocessing of text can be done.

Next, we have to build a model by implementing RNN for the data which is best suitable algorithm which gives the prediction accurately. And developed it to use as a real time prediction problem for grading the essay.

There are huge number of phases in automated essay grading based on Recurrent Neural Network.Data collection is the first phase, by this phase data should be collected not usually a less data set, it should be moderate data set according to the requirements one should collect or create the data for the detection. Text Preprocessing is done by implementing Natural Language Processing and this contain a lot of sub-phases for the processing of the text, it includes importing libraries, removing punctuations and numbers, converting each word into its lower case, Stemming, splitting data into train and text of the model.

Now we trained the model with Recurrent Neural Networks-LSTM. Our model gets saved in the backend of TensorFlow. Then by using this saved model we have written a code for prediction. Our model has predicts the grade of the essay given by the user . But here we need to copy the path and need to be paste in the code. It will be a problem for copying the path again and again. To overcome this problem, we use flask an awesome tool for model deployment. Flask is a web application framework written in python. It is very easy to make APIs and develop webpage in UI.

By using flask web application, we have created a webpage in UI and also HTML code is written to structure a webpage and its content.

Now in the UI, we have to give the input (Essay). The model takes the input and predicts the grade of the given essay.