A MAJOR PROJECT REPORT

ON

Examene - Design and development of an digital assessment systems for virtual examination

Enrollment No. : 9917103126, 9917103251, 9917103254.

Name : Lalit Garg, Abhinav Verma, Neha Agarwal.

Supervisor : Dr. Raju Pal.

December-2020

Submitted in partial fulfillment of the Degree of Bachelor of Technology

in

Computer Science Engineering.

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING INFORMATION TECHNOLOGY JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

DECLARATION

We hereby declare that this submission is our own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: Noida.

Date: 06, December 2020.

Name: Neha Agarwal

Enrollment No. : 9917103254

Name: Lalit Garg

Enrollment No. : 9917103126

Name: Abhinav Verma

Enrollment No. : 9917103251

CERTIFICATE

This is to certify that the work titled **Examene - Design and development of an digital assessment systems for virtual examination** submitted by *Abhinav Verma, Neha Agarwal, Lalit Garg* on partial fulfillment for the award of degree of **Bachelor's Of Technology** of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor: ....

Name of Supervisor: Dr. Raju Pal

Designation: ....

Date: 06, December 2020.

ACKNOWLEDGEMENT

We would like to express our special gratitude to our esteemed college, Jaypee Institute of Information Technology 128, for giving us this wonderful opportunity of making the major project of 2020-21, under the valuable supervision, timely suggestions and inspiration offered by our most respectable project mentor Dr. Raju Pal. It was because of his continuous encouragement that this report reached its successful completion. We also place on record and warmly acknowledge the contribution of our panel coordinators. Last but not the least we express our sincere thanks to all our friends who have patiently extended their support for accomplishing this undertaking.

Name: Neha Agarwal

Enrollment No. : 9917103254

Name: Lalit Garg

Enrollment No.: 9917103126

Name: Abhinav Verma

Enrollment No.: 9917103251

SUMMARY

The projects title is ”Design and development of an digital assessment systems for virtual examination” in which we are trying to produce a solution for modern problem, as the world is moving towards the automation and digital age, so there is a need of automation in the Indian examination system as well. The current pandemic has raised many questions, has shown how badly Indian educational institutions’ is lacking in creating fair environment for student and teachers as well. Major problem is to get a fair platform to automatically generate question from a given syllabus, and checking of answers require a high concentration, time, and, energy which has increased work load of teachers immensely and are prone to mistakes as well. Hence there is a need for automatic system which can generate questions, check the answers and, generate the performance. Also, managing a huge database of question-answer systems and different unfair means like paper-leakage, and generating different sets of questions is a major concern. In the manual system, it may be possible that different marks are given for the same answer. Solving all these problems, this system can lead to more efficient, fair, and, smooth conduction of examination.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General Introduction

The projects title is "Design and development of an digital assessment systems for virtual examination" in which we are trying to produce a solution for modern problem, as the world is moving towards the automation and digital age, so there is a need of automation in the Indian examination system as well.The major project focuses more on the research for information of the recent work and solutions proposed to overcome the challenges in the automatic generation of question-answer for the development of an efficient digital assessment system for virtual examination.We have implemented a web-based application where virtual examination can be conducted in the portal.Examination provides option for the subjects as well as for the type of test student wants to start-objective or subjective.

1.2 Problem Statement

The current pandemic has raised many questions, has shown how badly Indian educational institutions' is lacking in creating fair environment for student and teachers as well. Major problem is to get a fair platform to automatically generate question from a given syllabus, and checking of answers require a high concentration, time, and, energy which has increased work load of teachers immensely and are prone to mistakes as well. Hence there is a need for automatic system which can generate questions, check the answers and, generate the performance. Also, managing a huge database of question-answer systems and different unfair means like paper-leakage, and generating different sets of questions is a major concern. In the manual system, it may be possible that different marks are given for the same answer. Solving all these problems, this system can lead to more efficient, fair, and, smooth conduction of examination.in the manual evaluation, it is possible that students with the same answer may have different marks as it depends on the teacher mindset who evaluated the answer sheet but with the automated virtual examination system, same answers will always be judged using the same criterion.

1.3 Significance/Novelty of the problem

A question-answer is an efficient way of information retrieval. The objective of the research is to reduce time consumption and minimizing the human intervention in manually generating objective and subjective questions and automatically generating the responses by matching the human answer with the correct answer. Many researchers have proposed strategies for automatic question-answer generation. After analyzing various reports, journals, and optimization techniques given in the research papers, we are able to conclude a final optimized model that can result in the required examination assessment system. By this system, evaluation error will be reduced, minimizing the human intervention, and resulting in a fair platform for both students and teachers.

We have started from the research on basic NLP pipeline tools using tokenization, lemmatization, stemming, structure tree parser, part of speech tagging and then found significant work on the improvement of part of speech tagging using optimization techniques like NGrams, Hidden Markov model, Viterbi algorithm, Rule-based POS tagging, transformation-based tagging, removing lexical and syntactical ambiguities in the texts while retrieving information using text summarization techniques like TF-IDF algorithm for paraphrasing, finding an optimized set of answers synonyms, UML diagram to remove an attachment and homonymy syntactic ambiguity. We have also focused on the optimization of evaluating subjective answers using various methodologies.

1.4 Emperical Study

There is much research carried to find an optimized algorithm for Question-Answer formulation from the text. An experiment regarding Answer Evaluation using Machine learning shows that the proposed system can evaluate 5760 answers in a day whereas a human working for 8 hours can evaluate 480 answers a day. The scores are calculated for 10 students. The difference between manual evaluation and system evaluation is very close. The goal of the Deep Descriptive Answer Scoring model is to replace a system where marking and evaluation depends on factors like human mindset, time, thinking style, etc. The results predicted in the The model was compared with the previous models and the method proposed here give better results thus improves accuracy. A study on Generating Multiple Choice Items for Medical Text: A pilot study shows that Experimentation during development showed that the model improves by 30% using the Key-term identification and Source clause selection.

1.5 Brief Description of the Solution Approach

The main goal of the framework is to evaluate the performance of the students using an automated evaluation system without human intervention.
We will accept data using the Web UI about the subject chosen for evaluation. Another input will be the type of test-objective or subjective test. Taking the username of the student, subject name, and the type of test category the system will automatically generate questions from the text which is already stored corresponding to the subject chosen in the file.
Procedure to generate Question-Answer:
NLP tools-tokenization, stemming and lemmatization is used to pre-process the text to remove unwanted and stop words. The tf-IDF algorithm is used to summarize the text and synonym paraphrasing is used to generate a slightly variated version of the

text for mixing hard level questions with easy questions. The sentence is tokenized into words. After Part of Speech tagging, grammar rules, and Stanford parser, sentences with correct given grammatically sequence is filtered.

For the Objective test, key-term is identified using POS tagging. The key-term is blank out to generate questions and a list of synonyms with respect to the answer is generated. The question paired with the correct answer and a list of synonyms is stored. For the Subjective test, the question is phrased by blanking the term. Here the semantic analysis is used to ensure the correct wh type of question is generated. Features like the number of keywords, length, average word length are extracted from the user answer.LSTM is used to train the model to learn the relation between the text sequence of user answers. After learning from LTM and matching the extracted features of the user answer with the actual answer, the score is calculated predicting the accuracy of the user-answer. The score reveals the extent of similarity of user answers with the actual answer. Synonyms Paraphrasing is used to check all the relevant words that can be replaced by the words in the exact answer.

1.6 Comparision of existing approaches to the problem framed

Existing approach of UKDiss ALgorithm to generate Question from the text overview the NLP tools to generate the algorithm for question-answer. The training step follows these steps: Load the text, Detect potential sentences which can generate questions, Mark important terms which can be marked as blank, Using grammar and parsing, mark your grammar rule and filter the sentences according to the given grammar and Store the questions and respective answers for evaluation but several lexical and syntactical challenges were faced by the existing approaches. For example, Paraphrasing is a challenge where text should be a slightly variated version of the text to avoid direct straightforward questions. Negated Sentences and the ones starting with pronouns should be avoided, POS Ambiguity where one token can have different tags of speech.

A study on generating Multiple Choice Test from a Medical text highlights that Multiple Choice Question is the most common way of educational assessment but it is a very time-consuming task for teachers. In order to save time and labor, Mitkov proposed a system in 2006 which extracts the important key term from the syllabus and automatically generates multiple-choice questions for assessment. Mitkov has proposed a three-step methodology-

1. Sentence Parsing
2. Key-term identification
3. Source Clause Selection

The rules of Source Clause Selection were that negated sentences and the one starting with the pronouns should not be used to generate questions which resolves one of the shortcomings of the previous approach discussed.

A study on the answer evaluation using machine learning says that Manual answer evaluation is a time-consuming task and requires a lot of human intervention.Also, there is less chances of fair marks given to same answers.Thus the system proposed here will first tokenize the answer in words and then evaluate answers using machine learning tools performing automatic evaluation of answer.Only one has to scan and give answer sheet to the system, the algorithm will extract the suitable features from the user provided answer to match it with the correct answer.Such system is required to provide eligible and fair marks.This will reduce the time and cost of institutions that they pay for manual evaluations.
The algorithm assigns marks on basis of :
The Number of keywords matched and the Length of the answers.
The proposed system can evaluate 5760 answers in a day whereas a human working for 8 hours can evaluate 480 answers a day. The scores are calculated for 10 students.The difference between manual evaluation and system evaluation is very close.

Chapter 2

Literature Survey

2.1 Summary of papers studied

The report describes the research for information on the recent work and solutions proposed to overcome the challenges in the automatic generation of question-answer for the development of an efficient digital assessment system for virtual examination. A question-answer is an efficient way of information retrieval. The objective of the research is to reduce time consumption and manpower in manually generating objective and subjective questions and automatically generating the responses by matching the human answer with the correct answer. Many researchers have proposed strategies for an automatic question-answer generation. After analyzing all the findings and optimization techniques in the research papers, we are able to conclude a final optimized examination assessment system. By this system, evaluation error in the marks will be reduced. We have started from the research on basic NLP pipeline tools using tokenization, lemmatization, stemming, structure tree parser, part of speech tagging and then found significant work on the improvement of part of speech tagging using optimization techniques like hidden Markov model (Viterbi Algorithm), Rule-based POS tagging, transformation-based tagging, removing lexical and syntactical ambiguities in the texts while retrieving information using text summarization techniques like TF-IDF algorithm for paraphrasing, finding an optimized set of answers synonyms, UML diagram to remove an attachment and homonymy syntactic ambiguity. We have also focused on the optimization of evaluating subjective answers using various methodologies.

2.1.1 Algorithm for generating questions from the text

[11]A framework has been proposed to evaluate the performance of students. This is done to check whether a given text has been read carefully by the student. The methodology describes the fill in the blanks, true-false generation questions using multiple tools of nlp preprocessing. It is divided in two phases, first is the training phase where the given text syllabus is trained and preprocessed to generate questions and another is the evaluation phase where student-generated answers are matched with the already stored answer.

The first training phase follows these steps:

1. Load the text.
2. Detect potential sentences which can generate questions.
3. Mark important terms which can be marked as blank.
4. Using grammar and parsing, mark your grammar rule and filter the sentences according to the given grammar.
5. Store the questions and respective answers for evaluation.

2.1.2 Generating Multiple Choice test from a Medical Text: A Pilot Study

[5]Multiple Choice Question is the most common way of educational assessment but it is a very time-consuming task for teachers. In order to save time and labor, Mitkov proposed a system in 2006 which extracts the important key term from the syllabus and automatically generates multiple choice questions for assessment. Mitkov has proposed a four step methodology-

1. Sentence Parsing
2. Key-term identification
3. Source Clause Selection
4. Transformation to stem

After sentence-parsing using grammar, Mitkov highlighted the identification of key-term for generating questions.Multiple Choice test items should have key-term as its anchor rather than other information.

2.1.3 Single Document Automatic Text Summarization Using Term Frequency - Inverse Document Frequency (TF-IDF)

[3]Text summarization removes less important works and reduces to its much simpler version and thus helps the reader to extract the important information out of text. This research aimed to implement TF-IDF algorithm and manges to compare it with other text-summarization algorithms. The methodology of the method used in the approach produces 67 of accuracy with three data samples which are higher compared to other text-summarization algorithms.

The algorithm uses existing libraries NLTK and TextBlob.

Tf-IDf is a numerical calculated value for each word which measures the importance of each word with respect to the given text. Summary generation is the descending order of the words with one with higher TF-IDF value listed before.

TF IDF value indicates the following importance with reference to the text.

TF=total count of words in a document/total number of words in a document

IDF=log (All document Number/Document frequency)

Preprocessing of the algorithm comprises part of speech tagging, word stemming and removing stop words like a ,an, the etc. Then Feature extraction is done by obtaining the sentences from a document and allotting a value between zero and one.Last, summary generation is the sentences that are put into the summary in the order of the position in the original document.

Load text document → Text preprocessing → Calculate TF-IDF value → Calculate each sentence score → Summary generation

2.1.4 A Study on Different Part of Speech(POS) Tagging Approaches in Assamese Language

[10]Part of speech tagging plays a very crucial role in Natural language processing. It has an international and worldwide approach. It is called POS tagging. POS tagging is a piece of software that pairs up each word to its speech grammatical category-pronoun, noun, adverb, verb, adjective and thus it is a crucial part in natural language processing algorithms.

Architecture of POS tagger:

1. Tokenization

The sentence in the text is divided into words(tokens). The tokens may be words,punctuation marks.

2. Ambiguity look-up

Lexicon and guessor is used for unknown words to predict the part of speech.A lexical analyzer is a program which breaks a text into lexemes(tokens).

3. Ambiguity Resolution

If an expression(word/phrase/sentence) has more than one POS tag associated with it is, then it is referred to as ambiguous.

For example: She saw a bear.

Your efforts will bear fruit.

The word bear in the above sentences has completely different senses,but more importantly one is a noun and other is a verb.

POS Tagging Techniques:

1. Rule-based POS Tagging

It is the oldest method which lists hand written rules for tagging to identify the correct tag in case of ambiguity.

2. Markov Model

A finite state machine in which each state has two probability distributions.The probability of emitting a symbol and probability of moving to a particular state.The aim of Markov Model is to find optimal sequence of tags T=t1,t2,t3,... for a proper word sequence W=w1,w2,w3...Here probability of future state depends on the previous state.

3. Hidden Markov Model

It is a dynamic approach of Markov Model with improveriation in Emission and transition probability table.It is also known as Viterbi Algorithm. It is called Hidden Markov Models because state transitions are not observable.

2.1.5 Automatic Generation of Assessment Test Items from Text: Some Quality Aspects

[6]Quality aspects of Sequential approach of text processing is listed here:

1. Text preprocessing

It is the conversion from a raw text file into a well-defined sequence of linguistically correct texts.It consist of two processes:document triage and text segmentation.In Document trauge, document is converted from a digital file into a well-defined text document.It is accomplished using software tools.This stage is completely technical.

2. Segment filtering

Every text sentence is not certain to be a potential sentence to frame questions.For qualitative automated evaluation,proper filtering of acquired sentences should be considered so that it has the most salient segments.Various features like sentence-length cut-off(short sentences are excluded),use of cue phrases (inclusion of sentences with phrases such as "in conclusion"), sentence position in a document/paragraph, occurrence of frequent terms (based on TF-IDF term weighting), and occurrence of title words.

3. Test-item generation

The main problem here is to which word to blank out to generate questions.Here blanking out special items rather than common words is given much importance.

Example: Source: Ram has made a beautiful scenery with a perfect blend of colours using crayons.

Result: Ram has made a beautiful —— (what?) with a perfect blend of colours using crayons.

Another issue, which arises at this step, is that the processed sentences may contain anaphora. Without an implementation of automatic anaphora resolution,the user could resolve the anaphora manually (e.g. to replace pronouns with corresponding nouns) using the in-context display of the processed sentence.

2.1.6 Generating Natural Language Questions to Support Learning ON-Line

[7]Regardless of the method used, the procedure must perform four phases:

1. Content-selection: selecting spans of source text from which questions can be generated.

2. target-identification: determining which specific words and phrases should be asked about.

3. Question-formation: identifying the suitable question from the content selected.

4. Surface form generation: producing the final surface-form realization.

The main method here used is semantic based using the techniques of semantic role labeling(SRL). Semantic role labeler identifies the semantic entities associated with each predicate.Set of SRL includes ArgM-LOC (location), ArgM-EXT (ex-tent), ArgM-DIS (discourse), ArgM-ADV (adver-bial), ArgM-NEG (negation), ArgM-MOD (modal verb), ArgM-CAU (cause), ArgM-TMP (time), ArgM-PNC (purpose), ArgM-MNR (manner), and ArgM-DIR (direction).

After the process of tokenization and POS tagging, a single word is first paired with a semantic role label to ensure the semantic correctness of the question generated. For example, AM-LOC can be used to generate a where question,an AM-TMP can be framed as when question,AM-PNC can be framed as a why question, AM-MNR can be framed as How question.

2.1.7 Resolving Syntactic Ambiguities in Natural Language Specification of Constraints

[2]Stanford POS tagger and parser is used to tag the English test and is capable of 0.90 accuracy.Accuracy of the Stanford parser is low for real-time applications.

The two types of Syntactic ambiguity are:

1. Attachment Ambiguity In the example: The pay is given to all the employees with bonus,the ambiguity is due to the prep with phase.There can be two meanings of this same text. One meaning can be that pay is given to those employees who got bonus. Another meaning can be that pay with bonus is given to all employees.Thus prep with phase can have two meanings depending on its attachment.

2. Homonymy

It is a type of syntactic ambiguity in which a token in a sentence can exhibit one or more tag of speech and thus making the process of POS tagging ambiguous.

2.1.8 Answer Evaluation using Machine Learning

[1]Manual answer evaluation is a time-consuming task and requires a lot of human intervention.Also, there is less chances of fair marks given to same answers.Thus the system proposed here will first tokenize the answer in words and then evaluate answers using machine learning tools performing automatic evaluation of answer.Only one has to scan and give answer sheet to the system, the algorithm will extract the suitable features from the user provided answer to match it with the correct answer.Such system is required to provide eligible and fair marks.This will reduce the time and cost of institutions that they pay for manual evaluations

The proposed system can evaluate 5760 answers in a day whereas a human working for 8 hours can evaluate 480 answers a day. The scores are calculated for 10 students.The difference between manual evaluation and system evaluation is very close.

2.1.9 Grading Descriptive Answer Scripts using Deep Learning

[12]The model is a combination of NLP and Machine learning.Using LSTM-Recurrent neural networks, the model learn the relationship in the text. LSTM was developed to overcome the shortcomings of Recurrent Neural Network. Deep Descriptive Answer Scoring model is used to predict the answers.

Long Short Term Memory(LSTM) is an artificial recurrent neural network architecture.Unlike standard deep Learning models, LSTM has feedback connections.It has operations to remember or forget information.It can evaluate entire sequence of input data.LSTM are advanced versions of Recurrent neural networks.

Different steps in the proposed model is listed:

1. Preprocessing Preprocessing phase takes the relevant features of the text and converts it into vectors.Using tokenization and Lemmatization, features are extracted.The score corresponding to the feature is represented as a one-hot vector.

2. Sentence Embedding

The embedding layer simply converts the feature corresponding to a word into glove vectors.The main aim of Using LSTM-RNN is to create a simplified and low-dimension semantic values by sequentially and recurrently Processing and finding the corresponding values for each token in a sentence.

3. Grading

Based on the embedding vector of the LSTM-RNN made in the second phase of Sentence Embedding, the output Layer, fully connected neural network layer, the dense layer will predict the one-hotted score.Keras library is used here.

2.1.10 Automated Essay Scoring

[4]**1. Project Essay Grader(PEG)**

On the request of the College Board, Ellis Page in 1966 developed Project Essay Grader system. It was the first AES system to be built. Indirect measures of writing skills were considered thus it made the PES system less traceable. Along with several parsers and dictionaries, special collections and classification criteria were also used.

2. Intelligent Essay Scorer(IEA)

It analyzes and scores using a semantic text analysis method called Latent Semantic Analysis. A psychologist Thomas Landauer developed the LSA approach. Latent Semantic Analysis is a method that maps semantic similarity between the words in the text. In the LSA based approach, we input the text in the form of a matrix of size (number of words in the text) * (number of features/context with which we want to analyze each word). Row in the matrix represents each word and column in the matrix represents the context./feature.Value in the cell consists of the frequency of word corresponding to the row and then the value is re-evaluated considering the frequency of the word and its importance with respect to the context mapping the particular column.

2.1.11 Automatic Assessment of Descriptive Answers for [9]Online Examination using Semantic Analysis.

The phases of the method proposed:

1. To find a set of suitable relevant answers for all questions in the question set.

2. To identify the patterns of answers to match with the user answer.

3. Implement stemming, stop words removal, part of speech tagging-all natural language processing pipeline tools to preprocess the text. Once unwanted words are removed from the user answer, they are prepared to feed into a neural layer as a matrix.

4. Classifying the answer using the criterion of semantic weights. To check for other relevant words, a lexical resource WordNet is used.WordNet is a lexical database dictionary NLTP toolkit majorly used to extract the relevant word considering a certain meaning. Here Artificial Neural Network is used to find the relation between the text using the sigmoid function. Long-Short Term Memory Network is used here. Feedback connections in LSTM retain the memory of the pattern in the sequence of inputs and thus gives better results.LSTM is an advanced-technology version of Neural networks.

5. On the basis of length, the number of keywords, relation of the sentence formation, score for the answer of candidates can be formulated. Final results are displayed with parameters such as accuracy and false-positive ratio.

2.1.12 Extracting Word Synonyms from Text using Neural Approaches

[8]Word embeddings (also known as distributional word representations) are vector representations for words that are usually constructed from raw text based on linear context (words that occur in the neighborhood of a target word). In these repre-

sentations, each word is converted into a vector of numerical values or real values. Computational techniques to extract synonyms from the raw text have been inspired by the classical distributional hypothesis “if two words have almost identical environments, we say that they are synonyms”.It addressed the synonymy identification problem as a classification task. To make the problem simpler and doable within the time-frame of the research, only adjectives were considered for the classification. The method used a feed-forward neural network with backward propagation as a learning algorithm. To obtain labeled training data, we extracted synonyms pairs from the SimLex-999 similarity lexicon with a similarity score ¿ 6:5. SimLex-999 is a gold-standard resource for evaluating distributional semantic models.

2.2 Integrated summary of the literature studied

S.No	Name	Significance of work-Findings
1	Algorithm for generating questions from the Text	Natural text processing pipelines method Tokenization, Stemming, POS tagging, Structure tree Parser to generate a question-answer system from the text. Lexical and Syntactical Challenges were discussed like Paraphrasing, Pos tagging Ambiguity process, inefficiency to learn connections in the answer sentences.
2	Generating Multiple Choice tests from a Medical Text.	The method here gave us two key points to strengthen our NLP pipeline methods. Importance of answer in the fill in the blanks as a key term. Filtration of inappropriate structures for the main text chosen-Subordinate clause, Negated clause, Coordinated NP, initial pronouns.
3	Single Document Automatic Text Summarization Using Term Frequency-inverse Document Frequency (Tf-IDF)	Text summarization technique TF-IDF algorithm to overcome the challenge of paraphrasing(questions should have a slight variation from the original text) and dealing with complex text by first simplifying the text.

4	A Study on Different Part of Speech(POS) Tagging Approaches in Assamese Language	The technique of assigning an appropriate part of speech tag for each word in an input sentence of a language is called Part of Speech Tagging. It is commonly referred to as POS tagging. It is the main method in our naive process.
5	Automatic Generation of Assessment Test Items from Text: Some Quality Aspects	Using text-preprocessing, segment filtering, test item generation, extract question fragments from the text were extracted
6.	Generating Natural Language Questions to Support Learning ON-Line	Use of semantic-based methods to optimize framing question-segments. Given a sentence, a semantic role label identifies the predicates (relations and actions) along with the semantic entities associated with each predicate. A set of modifiers is also defined and includes ArgM-LOC (location), ArgM-EXT (ex-tent), ArgM-DIS (discourse), ArgM-ADV (adverbial), ArgM-NEG (negation), ArgM-MOD (modal verb), ArgM-CAU (cause), ArgM-TMP (time),ArgM-PNC (purpose), ArgM-MNR (manner), and ArgM-DIR (direction).

7	Resolving Syntactic Ambiguities in Natural Language Specification of Constraints	Types of Syntactic Ambiguity faced:1. Attachment Ambiguity 2. Homonymy Ambiguity Resolution of Ambiguities: As attachment ambiguity is due to the ambiguous role of nouns with a preposition in a sentence. To correctly identify the attachment of the noun with the other two nouns, we map the (three) candidate English elements (such as nouns) to the classes in the UML class model. If the token matches to a relationship name then it is a verb or if the ambiguous token matches to a class-name or attribute-name then it is classified as a noun
8	Answer Evaluation using Machine Learning	The algorithm will evaluate the answer based on the length of the answer and important keywords covered which are specified by the teacher with each answer which is to be evaluated. The proposed system can evaluate 5760 answers in a day whereas a human working for 8 hours can evaluate 480 answers a day

9	Grading Descriptive Answer scripts using Deep Learning	It is a combination of NLP and machine learning. The learning is done by using the Recurrent Neural Network and LSTM cells. Deep Neural networks are able to capture the semantics of text in order to and the similarity between texts.
10	Automated Essay Scoring	Automated Essay Scoring Systems- Project Essay Grader uses average word length, essay length, number of semicolons, or commas to predict the intrinsic quality of essays. Intelligent Essay Assessor analyzes and scores an essay using a semantic text analysis method called Latent Semantic Analysis (LSA) that uses the semantic similarity between texts for evaluation
11	Automatic Assessment of Descriptive Answers for Online Examination using Semantic Analysis	Answers of candidates can be ranked based on measures of distance between keywords, numbers of keywords matched, and other similar heuristic metrics. This model will classify all answers based on similarity weights. Using ANN evaluation of the marks is done automatically according to the current weights (range will be 0.01 to 0.99).

12	Extracting Word Synonyms from Text using Neural Approaches	Neural techniques (such as Word2Vec) have been recently utilized to produce distributional word representations (also known as word embeddings) that capture semantic similarity/relatedness between words based on linear context.

Table 2.1: Integrated Summary

Chapter 3

Requirement Analysis and Solution Approach

3.1 Overall description of the project

A question-answer is an efficient way of information retrieval. The objective of the research is to reduce time consumption and minimizing the human intervention in manually generating objective and subjective questions and automatically generating the responses by matching the human answer with the correct answer. Many researchers have proposed strategies for automatic question-answer generation. After analyzing various reports, journals, and optimization techniques given in the research papers, we are able to conclude a final optimized model that can result in the required examination assessment system. By this system, evaluation error will be reduced, minimizing the human intervention, and resulting in a fair platform for both students and teachers.The current pandemic has raised many questions, has shown how badly Indian educational institutions' is lacking in creating fair environment for student and teachers as well. Major problem is to get a fair platform to automatically generate question from a given syllabus, and checking of answers require a high concentration, time, and, energy which has increased work load of teachers immensely and are prone to mistakes as well. Hence there is a need for automatic system which can generate questions, check the answers and, generate the performance.

3.2 Requirement Analysis

Functional Requirements:

1. Username of the student
2. Subject name
3. Type of the test-Objective/Subjective

Non-Functional Requirements:

1. The processing of each request should be done within 10 seconds
2. Site should load in 5 seconds.

3.3 Solution Approach

The main goal of the framework is to evaluate the performance of the students using an automated evaluation system without human intervention.
We will accept data using the Web UI about the subject chosen for evaluation. Another input will be the type of test-objective or subjective test. Taking the username of the student, subject name, and the type of test category the system will automatically generate questions from the text which is already stored corresponding to the subject chosen in the file.

NLP tools-tokenization, stemming and lemmatization is used to pre-process the text to remove unwanted and stop words. The tf-IDF algorithm is used to summarize the text and synonym paraphrasing is used to generate a slightly variated version of the text for mixing hard level questions with easy questions. The sentence is tokenized into words. After Part of Speech tagging, grammar rules, and Stanford parser, sentences with correct given grammatically sequence is filtered.
For the Objective test, key-term is identified using POS tagging. The key-term is blank out to generate questions and a list of synonyms with respect to the answer is generated. The question paired with the correct answer and a list of synonyms is

stored. For the Subjective test, the question is phrased by blanking the term. Here the semantic analysis is used to ensure the correct wh type of question is generated. Features like the number of keywords, length, average word length are extracted from the user answer.LSTM is used to train the model to learn the relation between the text sequence of user answers. After learning from LTM and matching the extracted features of the user answer with the actual answer, the score is calculated predicting the accuracy of the user-answer. The score reveals the extent of similarity of user answers with the actual answer. Synonyms Paraphrasing is used to check all the relevant words that can be replaced by the words in the exact answer.

Chapter 4

Modeling and Implementation Details

The projects title is ”Design and development of an digital assessment systems for virtual examination” in which we are trying to produce a solution for modern problem, as the world is moving towards the automation and digital age, so there is a need of automation in the Indian examination system as well. The current pandemic has raised many questions, has shown how badly Indian educational institutions’ is lacking in creating fair environment for student and teachers as well. Major problem is to get a fair platform to automatically generate question from a given syllabus, and checking of answers require a high concentration, time, and, energy which has increased work load of teachers immensely and are prone to mistakes as well. Hence there is a need for automatic system which can generate questions, check the answers and, generate the performance. Also, managing a huge database of question-answer systems and different unfair means like paper-leakage, and generating different sets of questions is a major concern. In the manual system, it may be possible that different marks are given for the same answer. Solving all these problems, this system can lead to more efficient, fair, and, smooth conduction of examination..

4.1 Design Diagrams

4.1.1 Use Case diagrams

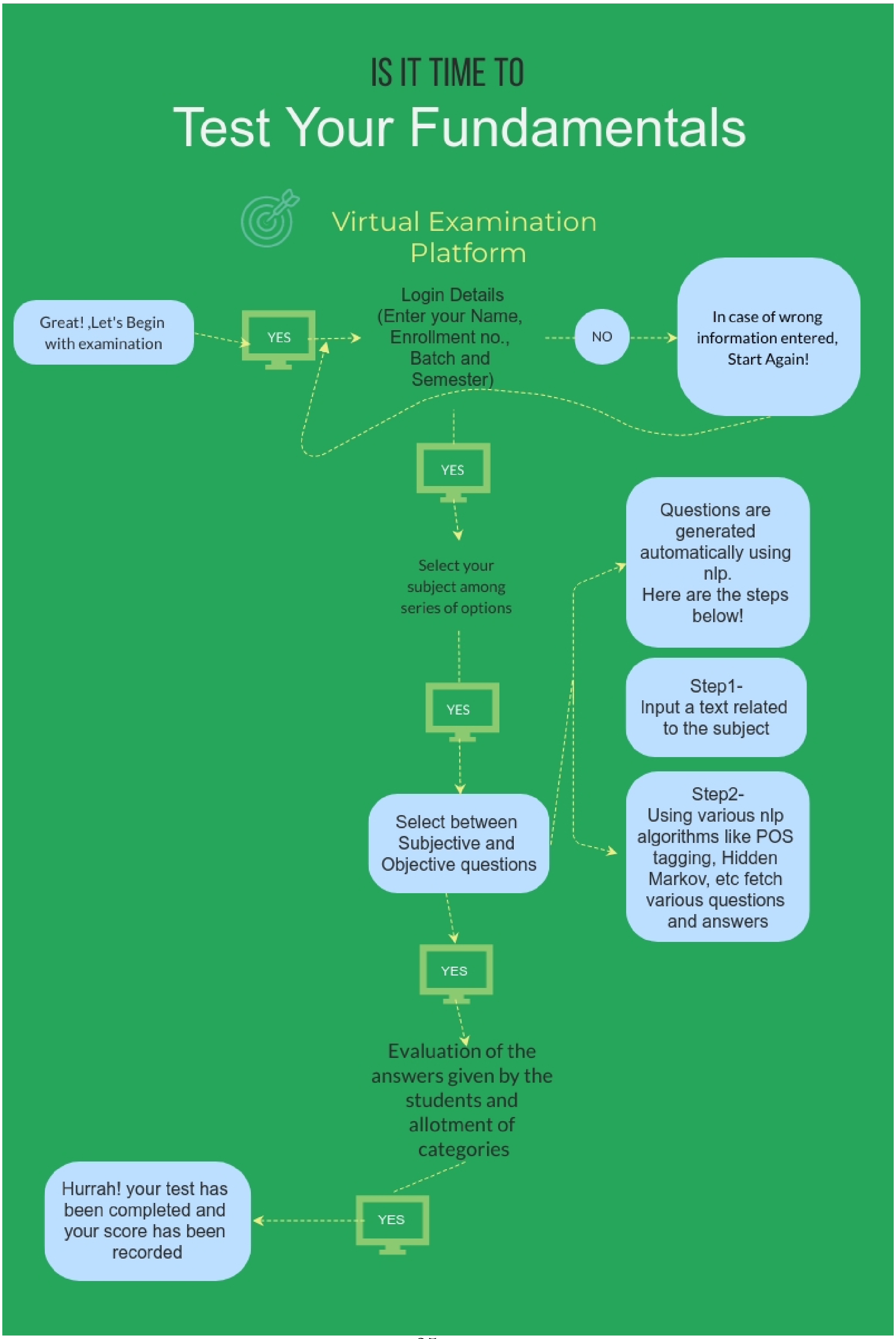

Figure 4.1: Flowchart of complete model.

4.1.2 Control Flow Diagrams

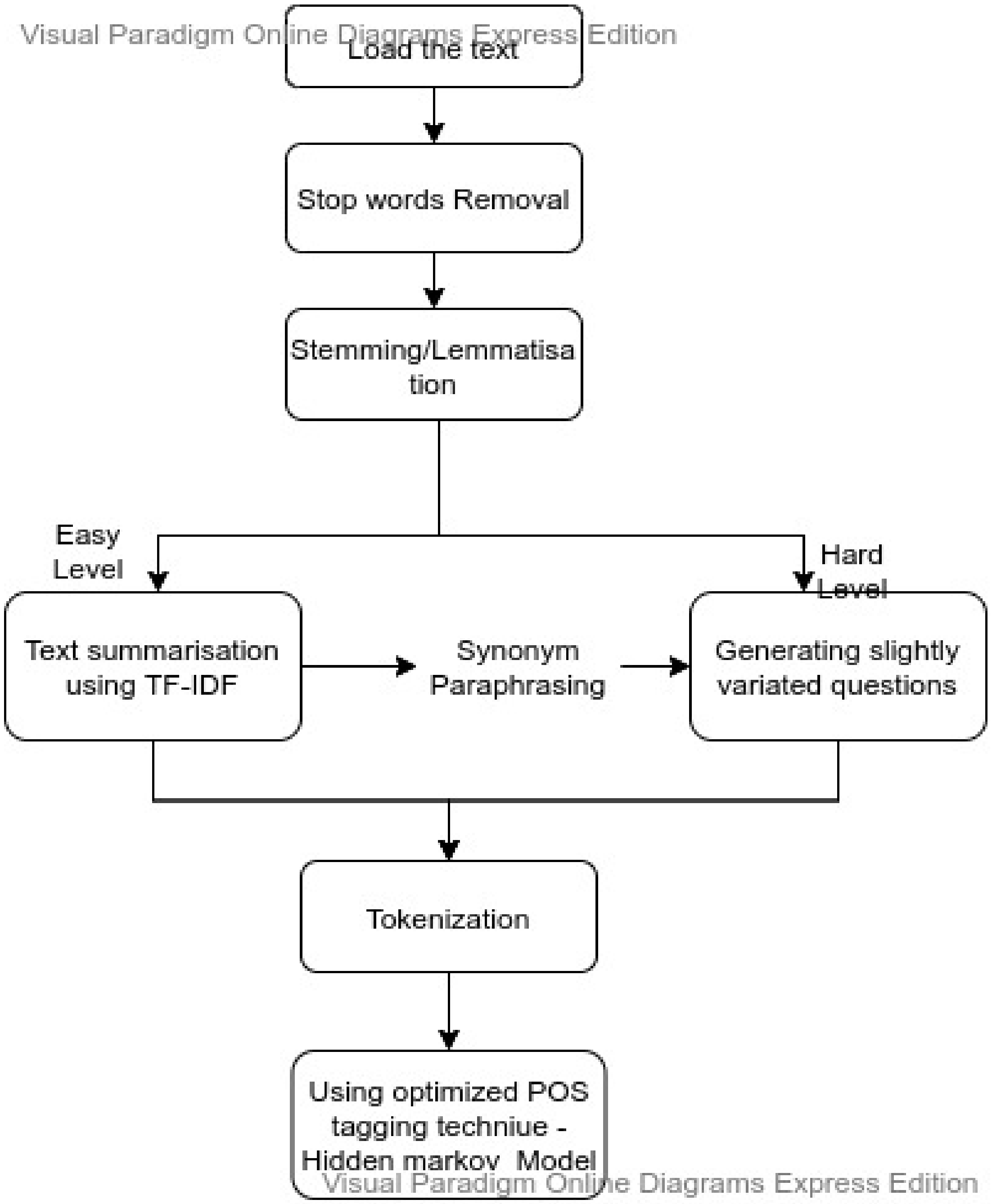

Figure 4.2: Workflow part one

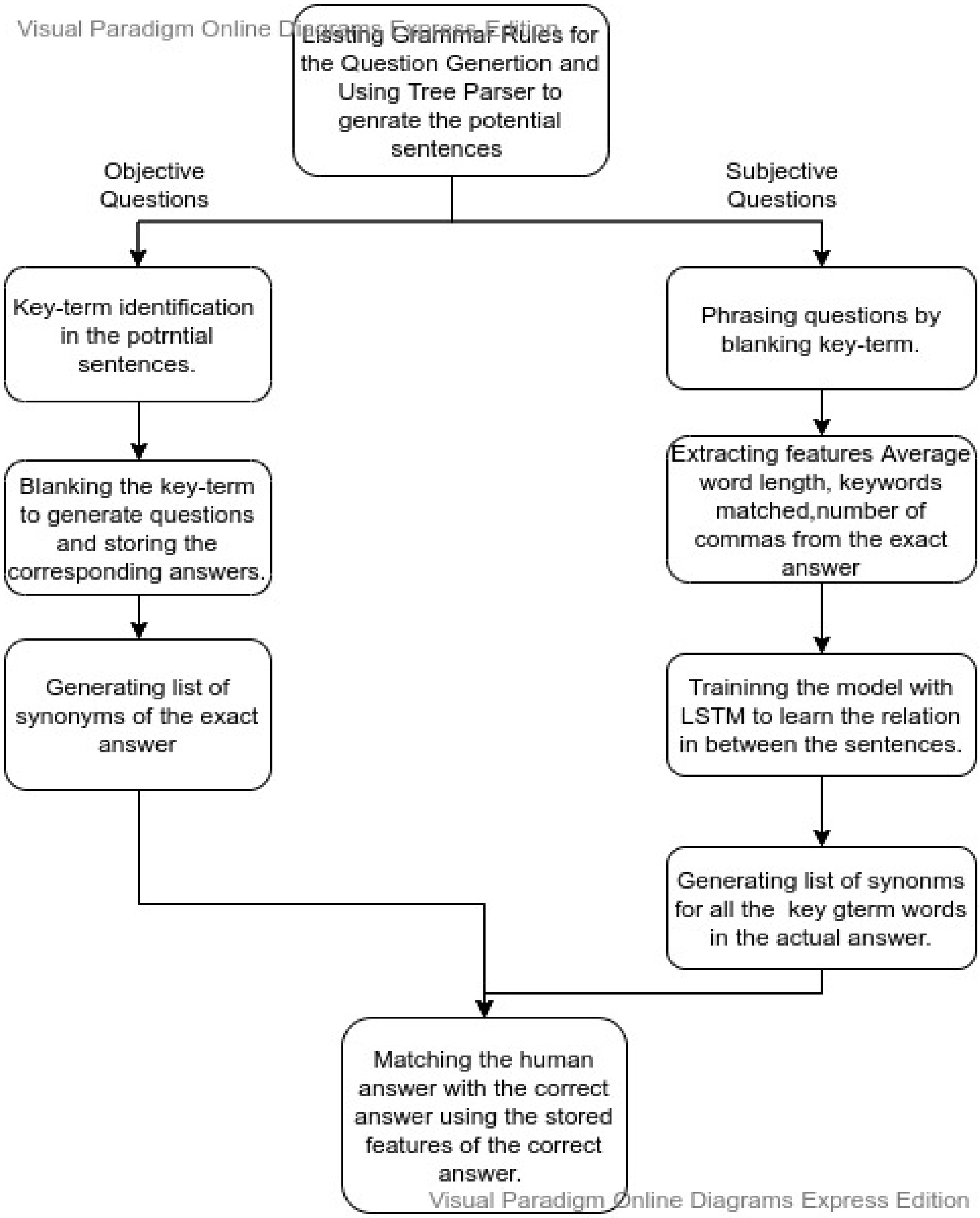

Figure 4.3: Workflow part two

4.2 Implementation details and issues

The main goal of the framework is evaluate the performance of the students, generate questions as well as try to rank the questions based on semantic correctness and difficulty level.

We will accept data using the Web UI in the form of text or doc file and will use it for the further process. Applying stemming on the input data. Then we will use Tf-Idf algorithm for summarising the content on the basis of the difficulty level selected chosen by the examiner. On the summarized data we will apply paraphrasing using wordNet and nlp, which makes the data more diverse to create question. The paraphrased data is tokenized and our main framework optimized POS tagging (Hidden Markov Model - Viterbi Algorithm) is applied on it, which create the lists on the basis noun, verb, models etc. Then we apply grammar rule on the above data obtained and create question for the test. Relation between words is analyzed using lstm-rnn and few other neural networks weather they are co-related to each other or not. After this correctness of the answer in calculated using nlp and neural network and score in generated on the basis of number of keyword match and length of answer.

Steps for the implementation is are as follow:

1. Load the text, remove the stop words and apply the stemming.
2. Generation of Question on the basis of Difficulty level selected.
3. Paraphrasing.
4. Tokenization.
5. Using optimized POS technique.
6. Applying grammar rule and creating questions.
7. Assessing answer using LSTM.
8. Calculating score.

4.2.1 Loading Data

We will accept data from the Web UI in the form of text or doc file and will use it for the further process. Applying stemming on the input data. Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. A stemming algorithm reduces the words "chocolates", "chocolatey", "choco" to the root word, "chocolate" and "retrieval", "retrieved", "retrieves" reduce to the stem "retrieve". Stemming is an important part of the pipelining process in Natural language processing. The input to the stemmer is tokenized words.

4.2.2 Generation of summary of the content on the basis of Difficulty level selected

On the basis of the level set by examination authorities, for easy level, text summarisation technique using TF-IDF algorithm will be executed. TF-IDF algorithm is used to extract the features of the document by obtaining the sentence in a text document based on its importance and given the value between zero and one.Summary generation is the sentences that are put into the summary in the order of the position in the original document.For hard level, sin order to generate slightly variated question, synonym paraphrasing of summarised text will be taken in account to generate hard-level questions.

4.2.3 Paraphrasing

Synonymous paraphrasing of a text-based on WordNet synonymy data and Internet statistics of stable word combinations (collocations). Given a text, we look for words or expressions in it for which WordNet provides synonyms and substitutes them with such synonyms only if the latter form valid collocations with the surrounding words according to the statistics gathered from the Internet.

Synonymous paraphrasing (SP) is such a change of natural language (NL) text or of its fragments that preserves the meaning of the text as a whole. Nearly every plain text admits SP (in contrast to lists of names, numerical data, poetry, and the like). Computational linguistics has always considered SP an important and difficult problem. In this paper, a method of local SP of NL texts based on WordNet synonymy information (synsets) and Internet-based statistics on stable word combinations (collocations) is proposed. To paraphrase a text, we look for words or multi-words in it that are members of a WordNet synset and substitute them with other members of the same synset only if they are feasible components of collocations with the surrounding words according to statistical evaluation through the Internet search engine, such as Google.

Various Types of Paraphrasing:

Text compression For this, the shortest synonym is taken in each synset (either independently of any statistical evaluations or selecting from the words that passed the marginality threshold). This gives a significant gain in space only when there are abbreviations (s) among absolute synonyms.

Text canonization For this, the most frequently used synonym is taken. Of course, it may prove to be the same one as in the source text.t is also useful for persons with limited language knowledge, i.e. for foreigners or children, since this renders texts in a more intelligible way.

Text simplification Any text will be more intelligible for a language-impaired person if we select among synonyms a "simpler". It is not always the most frequently used synonym.

4.2.4 Tokenization

Tokenization is the process of tokenizing or splitting a string, text into a list of tokens. One can think of token as parts like a word is a token in a sentence, and a sentence is a token in a paragraph.

4.2.5 Optimized PoS Tagging-Hidden Markov Model

Syntactic parsing is a necessary task that is essential for Natural language processing methodology including Part of Speech (POS) tagger. For the development and enrichment of languages, part of speech tagging plays a very crucial role. Part of speech tagging, especially for the regional Indian languages can give an international and worldwide approach. For a regional language like Assamese which is Assam's official language, part of speech tagging has become very much essential for the overall flourishment of the language. The technique of assigning an appropriate part of speech tag for each word in an input sentence of a language is called Part of Speech Tagging. It is commonly referred to as POS tagging.

POS Ambiguity: If an expression (word/phrase/sentence) has more than one interpretation, we can refer to it as ambiguous. The process to remove the ambiguity of words in a given context is called disambiguation.

POS Optimized Techinque-Hidden Markov Model(Viterbi algorithm): The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states – called the Viterbi path – that results in a sequence of observed events, especially in the context of Markov information sources and hidden Markov models. In an HMM, we know only the probabilistic function of the state sequence. At the beginning of the tagging process, some initial tag probabilities are assigned to the HMM.

4.2.6 Applying grammar rule and creating questions.

Mitkov filtered the clause in the text focusing on the rules and listed inappropriate structures for MCTIG (key terms underlined).

1. **Subordinate clause** - Although chest pain is a problem.
2. **Negated clause** - Ram should not play outside.
3. **Coordinated NP** - Excessive tension causes mental illness and blood pressure problems.
4. **Initial Pronoun** - It associates with hypertension instead.

Parsing Parsing is another important aspect utilized in conjunction with part-of-speech tagging to identify and understand natural language sentences. With parsing, when given an input sentence and a grammar, it can be determined whether the grammar can generate the sentence. Parsing can be described, at least in this context, as "the process of analyzing a string of words to uncover its phrase structure, according to the rules of the grammar".

After taking care of above listed inappropriate structures and filtering the appropriate source clause, a finite main clause which contains an NP headed by a key term and functioning as a source or object with all the subordinate clauses provided that it does not contain the inappropriate structures listed are found eligible for the source clause.

4.2.7 Assessing the answers

In case of objective question, we will be using the list of synonyms of exact answers for checking the correctness of the question and calculation of the marks. But in case of subjective questions, in-order to understand the relation between the words present in the answer we used LSTM. The objective of this model is to extract the semantics to efficiently represent the text in answer scripts and develop a model from the key and evaluated answer scripts to grade non evaluated answer scripts using deep neural networks. It is a combination of NLP and machine learning. The learning is done by using the Recurrent Neural Network and LSTM cells. Deep Neural networks are able to capture the semantics of text in order to and the similarity between texts. The goal of the system is to replace the traditional human evaluation of the answer sheet that depends on several factors such as time, mindset, presentation style, and so on. The embedding vector from the LSTM layer will be the semantic representation of the answer. Based on this value the output layer, fully connected neural network layer, the dense layer will predict the one- hotted score. Supervised training is used for this sequential model. The neural networks are called from Keras library. The final layer will then predict the score. The proposed Deep Descriptive Answer Scoring model (D-DAS model) is a sequential model that consists of an embedding layer, LSTM-RNN layer, dropout layer, and dense layer. The dense layer gives the one-hot encoded score for each answer. The proposed system is an automated descriptive answer checking and grading application using deep learning. For semantics interpretation and grading of descriptive answers, natural language processing, and deep learning tools have been used.

4.2.8 Calculating score

Manual answer evaluation is a very tedious task. Manual checking is a very time-consuming process and also requires lots of manpower. Also, paper checkers are not able to give marks equally. So, our system will evaluate answers based on some keyword, and also manpower will be saved. Only one has to scan the paper then, based on the keyword in the answer the system will provide the marks to the question according to the dataset present. Also, By this system, the evaluation error of the marks to the particular question will be reduced. So, our system will evaluate answers based on some keyword, and also manpower will be saved. Only one has to scan the paper then the system will split the answer using OCR, based on the keyword in the answer the system will provide the marks to the question according to the dataset present, There is a need for such an application which will provide an easy evaluation of answer and can provide eligible marks. The algorithm will evaluate the answer based on the length of the answer and important keywords covered which are specified by the teacher with each answer which is to be evaluated.

The algorithm assigns marks on basis of :

The Number of keywords matched and the Length of the answers.

The proposed system can evaluate 5760 answers in a day whereas a human working for 8 hours can evaluate 480 answers a day. The scores are calculated for 10 students.The difference between manual evaluation and system evaluation is very close.

4.3 Risk Analysis and Mitigation

1. The web application can take a lot of time if a lot of multi-users access it at the same time.

2. The algorithm needs to be designed in such a way that it ensures the semantic correctness of the question generated.

3. Grammar rules and parsing has to be done carefully to ensure proper structure formation of the question.

4. The web UI needs to take input of the credential of a particular student.

5. Text from which questions will be extracted needs to be a plain test. Animated Slides or flow chart in the figure will generate an error.

6. Answers imputed by students need to be in plain text. Flow charts or diagrams are not an accepted format.

Chapter 5

Chapter-5 Testing (Focus on Quality of Robustness and Testing)

5.1 Testing Plan

Type of Test	Has it been performed	Explanation	Software Components
Requirement Testing	Yes	Requirement specification must contain all the requirements that are needed by our system	Manual work needs to plan out all the software requirements, the time needed to develop, technology to be used, etc.
Unit Testing	Yes	Testing technology using which individual modules are tested to determine if there are any issues, by the developer itself.	A manual check is required.
System Integration Testing	Yes	Testing where individual components are combined and tested as a group.	Compiling full code and testing it together.

Performance testing	Yes	Testing to evaluate the input where the best and most optimal output is yielded by the system.	Testing results ensure this.

Table 5.1: Testing Plan

5.2 Component decomposition and type of testing required

Sr. No.	List of modules which require testing	Type of test performed	Techniques used for writing test cases
1	Standalone program generation	Requirement testing, performance testing	White Box
2	Augmentation	System/System Integration Testing	White Box
3	Validation	Testing	Black Box
4	Comparision	Performance Testing	White Box

Table 5.2: Component decomposition and type of testing required

5.3 List all test cases

Sr. No.	Test Case	Expected Results	Actual Result	Status
1	Install prerequisites such as python, nltk	Successful installation should happen	Successfully Installed	Passed
2	Installing jupyter	Successful installation should happen	Successfully Installed	Passed
3	Running the algorithm to generate questions from a plain text	Test cases should give appropriate results	Successful	passed
4	Building web application using Flask	Successful build should happen	Successful	Passed

Table 5.3: List of all test cases

5.4 Error and Exception Handling

Sr. No.	Type of Error	Debugging Technique
1	Non Optimal Results	Dry Run and Optimize the Algorithm
2	Packages Not found	Re-installation of packages
3	Validation and Testing	Manual Debugging
4	Flask build failed	Manual Debugging

Table 5.4: Error and Exception Handling

5.5 Limitations of the solution

1. Our web application is not strong to support a huge number of student users at a time. The response will get delay due to large traffic.

2. Our evaluation system is not camera-proctored to ensure a fair evaluation and avoid cheating.

3. The system cannot evaluate handwritten solutions. Answers need to be typed to check its similarity with the exact answer.

4. We have here implemented only two types of test-Objective or Subjective.

5. We cannot add a flowchart or Image in our handwritten solutions. Only plain text will be evaluated by the system.

6. We cannot upload animated slides to frame questions. Data has to be plain text stored in the text file under the folder maintained for each subject.

Chapter 6

Findings,Conclusion and Future Work

6.1 Findings

```
import re
import nltk
import numpy as np
from nltk.corpus import wordnet as wn


class ObjectiveTest:

    def generate_test(self, num_of_questions=3):
        trivial_pair = self.get_trivial_sentences()
        question_answer = list()
        for que_ans_dict in trivial_pair:
            if que_ans_dict["Key"] > 3:
                question_answer.append(que_ans_dict)
            else:
                continue
        question = list()
        answer = list()
        while len(question) < num_of_questions:
            rand_num = np.random.randint(0, len(question_answer))
            if question_answer[rand_num]["Question"] not in question:
                question.append(question_answer[rand_num]["Question"])
                answer.append(question_answer[rand_num]["Answer"])
            else:
                continue
        return question, answer
```

```
In [13]: model=ObjectiveTest("./dbms.txt")

In [14]: model.generate_test()
Out[14]: (['___________________ are allowed to continue while reads on the snapshot are happening.',
           '_________ also stores metadata, which is data about data, to ease its own process.',
           'locks) are held by two or more connections that are each needed by the other connections so that they are stuck
         in an infinite ___________________ is called Deadlock.'],
          ['Database writes', 'DBMS', 'wait loop'])
```

Figure 6.1: Objective questions code snippet

```
In [8]: import numpy as np
        import nltk as nlp

        class SubjectiveTest:

            def __init__(self, filepath):
                # Question pattern
                self.question_pattern = [
                    "Explain in detail ",
                    "Define ",
                    "Write a short note on ",
                    "What do you mean by "
                ]
                # Grammar to chunk keywords
                self.grammar = r"""
                    CHUNK: {<NN>+<IN|DT>*<NN>+}
                    {<NN>+<IN|DT>*<NNP>+}
                    {<NNP>+<NNS>*}
                """
                self.filepath = filepath
                try:
                    with open(filepath, mode="r") as fp:
                        self.summary = fp.read()
                except FileNotFoundError as e:
                    print(e)

            @staticmethod
            def word_tokenizer(sequence):
                word_tokens = list()
                for sent in nlp.sent_tokenize(sequence):
                    for w in nlp.word_tokenize(sent):
                        word_tokens.append(w)
                return word_tokens

            @staticmethod
            def create_vector(answer_tokens, tokens):
                return np.array([1 if tok in answer_tokens else 0 for tok in tokens])

            @staticmethod
```

Figure 6.2: subjective questions code snippet

```
                question_answer = list()
                for _ in range(3):
                    rand_num = np.random.randint(0, len(keyword_list))
                    selected_key = keyword_list[rand_num]
                    answer = question_answer_dict[selected_key]
                    rand_num %= 4
                    question = self.question_pattern[rand_num] + selected_key + "."
                    question_answer.append({"Question": question, "Answer": answer})
                que = list()
                ans = list()
                while len(que) < num_of_questions:
                    rand_num = np.random.randint(0, len(question_answer))
                    if question_answer[rand_num]["Question"] not in que:
                        que.append(question_answer[rand_num]["Question"])
                        ans.append(question_answer[rand_num]["Answer"])
                    else:
                        continue
                return que, ans

            def evaluate_subjective_answer(self, original_answer, user_answer):
                score_obt = 0
                original_ans_list = self.word_tokenizer(original_answer)
                user_ans_list = self.word_tokenizer(user_answer)
                overall_list = original_ans_list + user_ans_list
                vector1 = self.create_vector(original_ans_list, overall_list)
                vector2 = self.create_vector(user_answer, overall_list)
                score_obt = self.cosine_similarity_score(vector1, vector2)
                return score_obt

In [9]: model=SubjectiveTest("./dbms.txt")

In [11]: model.generate_test()

Out[11]: (['Define APPLICATION SOFTWARE.', 'Write a short note on CONSISTENCY.'],
          ['The application software, or the user-interface, then accesses the database and presents that information in a w
         ay which is easy for the user to interpret and understand.',
           'ACID acronym stands for the properties maintained by standard database management systems, standing for Atomicit
         y, Consistency, Isolation, and Durability.'])
```

Figure 6.3: subjective questions output snippet

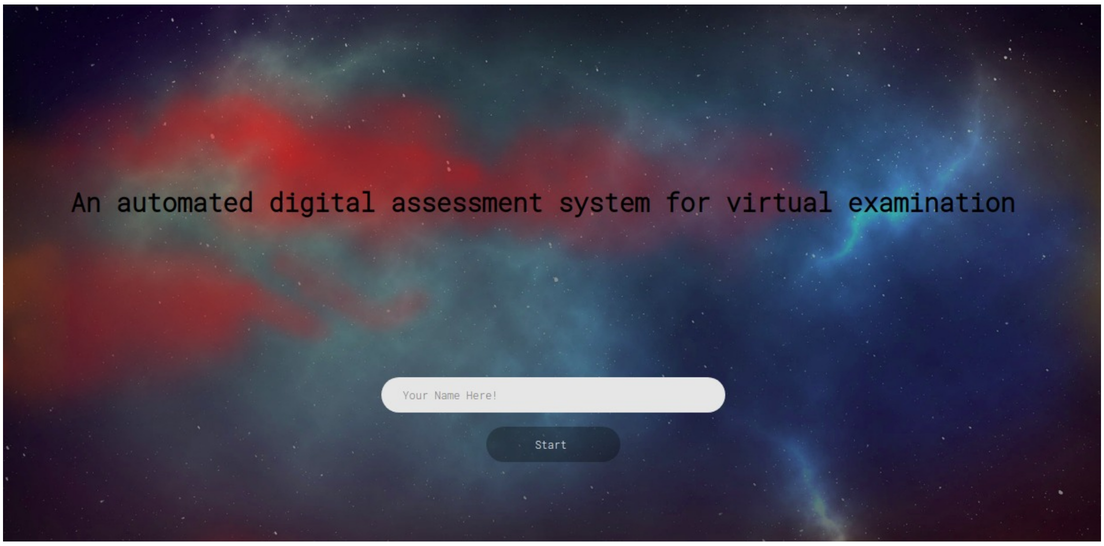

Figure 6.4: Web Interface

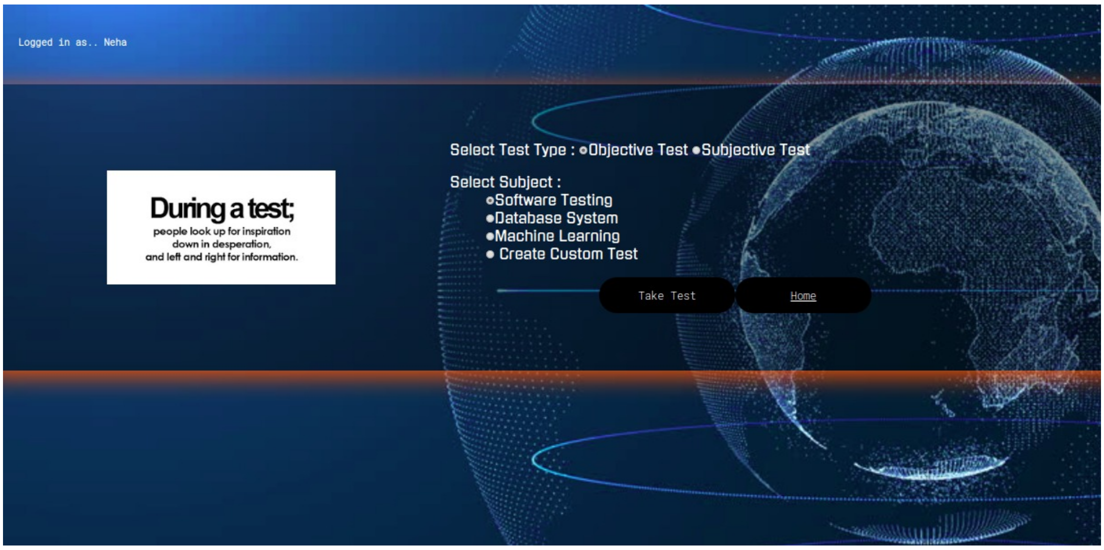

Figure 6.5: Login Page

Figure 6.6: One Word Answer type questions

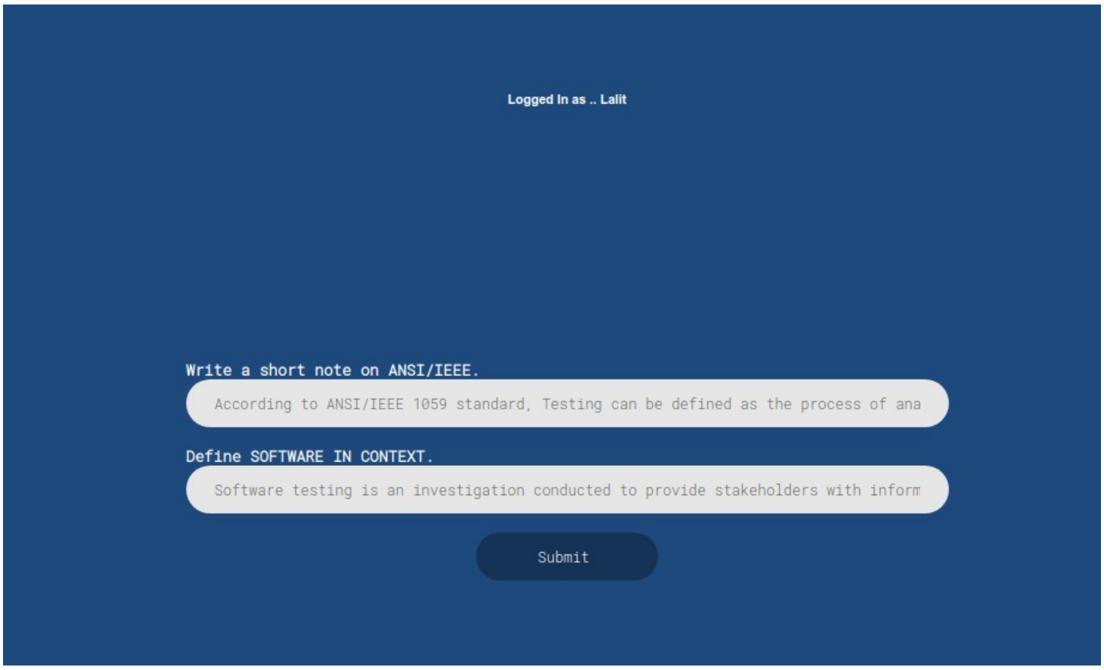

Figure 6.7: Subjective Answers type questions

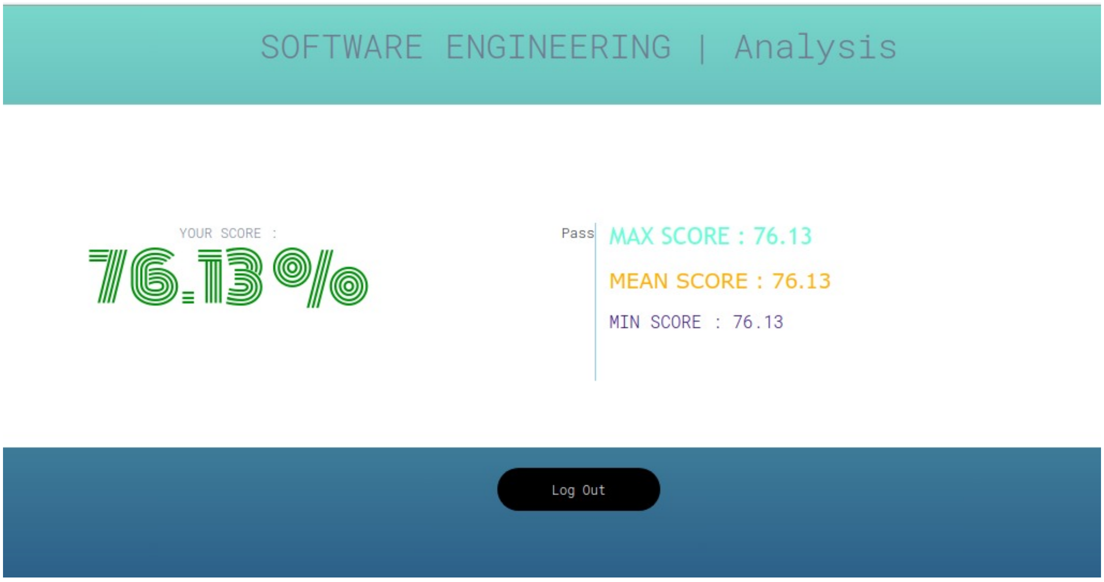

Figure 6.8: Final Score

6.2 Conclusion

In summary, the system to generate questions from text was data driven machine learning methods. The question generation process is a difficult activity in educational domain and is one of the research problems in modern days. The proposed system saves time, money and labor of the educational institutions. Automated evaluation ensures the fair and equal distribution of marks on the basis of features extracted using the proposed efficient algorithm. We have worked on the lexical and syntactical challenges mentioned in the Research paper and our algorithm achieves to overcome the challenges thus ensuring the lexical and semantic correctness of the questions framed.Synonyms Paraphrasing is used to ensure that Students who mug up the content without actual understanding will face variated questions of the original text.The proposed system is flexible, ease to use and applicable in different levels of education environments. Another application is it can be used as a self-evaluatory platform where student or adult can analyze his/her extent of reading and understanding skills. It is large scale ready with ability to learn more and more about different styles of topics and fields.

6.3 Future Work

1. The system can be extended in future to be able to recognize new statements with auto suggestion without needs to user intervention.

2. The system can be integrated with myCourses platform or can be leveraged and used with online learning portals like Coursera, Udacity.

3. Currently, a prototype will be implemented purely on python platform. However, this domain is open-ended.

4. This system can lead to more efficient, fair, and, smooth conduction of examination at large as well as small scale.

5. It can help in generating new type of questions altogether which is new for human thinking.

6. Another application is it can be used as a self-evaluatory platform where student or adult can analyze his/her extent of reading and understanding skills.

7. To avoid mass cheating in current times of pandemic, the system proposed where each student will get random number of questions on the basis of algorithm will be highly motivated.

Bibliography

[1] Tariq Ahmad, Lars H Lund, Pooja Rao, Rohit Ghosh, Prashant Warier, Benjamin Vaccaro, Ulf Dahlström, Christopher M O'connor, G Michael Felker, and Nihar R Desai. Machine learning methods improve prognostication, identify clinically distinct phenotypes, and detect heterogeneity in response to therapy in a large cohort of heart failure patients. *Journal of the American Heart Association*, 7(8):e008081, 2018.

[2] Imran Sarwar Bajwa, Mark Lee, and Behzad Bordbar. Resolving syntactic ambiguities in natural language specification of constraints. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 178–187. Springer, 2012.

[3] Hans Christian, Mikhael Pramodana Agus, and Derwin Suhartono. Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). *ComTech: Computer, Mathematics and Engineering Applications*, 7(4):285–294, 2016.

[4] Semire Dikli. Automated essay scoring. *Online Submission*, 7(1):49–62, 2006.

[5] Nikiforos Karamanis, Ruslan Mitkov, et al. Generating multiple-choice test items from medical text: A pilot study. In *Proceedings of the fourth international natural language generation conference*, pages 111–113, 2006.

[6] Andrey Kurtasov. Automated generation of assessment test items from text: Some quality aspects. In *AIST (Supplement)*, pages 91–95, 2014.

[7] David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. Generating nat-

ural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114, 2013.

[8] Nora Mohammed. Extracting word synonyms from text using neural approaches. *Int. Arab J. Inf. Technol.*, 17(1):45–51, 2020.

[9] Toshavi Patil. Automatic assessment of descriptive answers for online examination using semantic analysis. *Journal of the Gujarat Research Society*, 21(5):413–419, 2019.

[10] Bipul Roy and Bipul Syam Purkayastha. A study on different part of speech (pos) tagging approaches in assamese language. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(3), 2016.

[11] Koichi Sanoki and Ítalo Santiago Vega. Argumentative scheme in academic texts: Algorithm for generating questions on arguments. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6. IEEE, 2018.

[12] PJ Sijimol and Surekha Mariam Varghese. Handwritten short answer evaluation system (hsaes). *IJSRST*, 4, 2018.