KENYATTA UNIVERSITY

DEPARTMENT OF COMPUTING AND INFORMATION TECHNOLOGY

SCHOOL OF ENGINEERING AND TECHNOLOGY

Automated Examination Generation using Natural Language Processing and Artificial Neural Network

Peter Ndegwa Ndirangu

J57/39217/2017

"*This project report is submitted for the partial fulfillment of the requirements for the award of the degree of Masters of Science in Computer Science in the School of Engineering and Technology of Kenyatta University*."

APRIL 2023

DECLARATION

I declare that this report is my original work and has not been presented in any other university/institution for consideration of any certification. This project report has been complemented by referenced sources duly acknowledged. Where text, graphics, pictures, figures, or tables have been borrowed from other sources, including the internet, these are specifically accredited and references cited using the current APA system and following anti-plagiarism regulations.

Signature: -- Date …………………………

Name: Peter Ndegwa Ndirangu

Registration Number: J57/39217/2017

Department: Computing and Information Technology

Supervisor's Declaration

This project report has been submitted with my approval as a university supervisor.

Signature: ……………………………. Date………………………………

Name: Dr. Elizaphan Maina.

Department: Computing & Information Technology.

Kenyatta University.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

DEFINITION OF TERMS

ANN	Artificial Neural Network
NLP	Natural Language Processing
SVM	Support Vector Machine
LR	Logistic Regression
K-NN	K – Nearest Neighbor
MLA	Machine Learning Algorithms
DL	Deep Learning
IHL	Institutions of Higher Learning
API	Application Programming Interface
LMS	Learning Management System

ABSTRACT

The examination process is a key activity in evaluating what the learner has gained from the study. Institutions of Higher Learning (IHL) perform the activity by administering tests which comprises of questions and answers. Cognitive level, weight of the question, and topic coverage are key factors to consider when setting exams. The world today has largely focused on the automation of exam generation which has been ongoing with dire need during the period of the Covid-19 pandemic when education was greatly affected, leading to embracing online learning and examination. The process has taken shape; however, the automation process can be improved by incorporating machine learning algorithms in the process of setting examination. In view of this, the project focused on implementation of a question classification model that uses Neural-Network algorithm (NN) and Natural Language Processing (NLP) to determine questions cognitive levels based on the revised Bloom's Taxonomy. The iterative method of software development was adopted to provide room for continuous improvement. The developed model was put under test with a couple of questions obtained online. The effectiveness of the model was determined by subjecting it into database of 600 questions resulting to an accuracy of about 71%. An Application Programming Interface (API) and Moodle Learning Management System (LMS) plugin were consequently developed to allow integration of the model with an existing system. The deep learning approach was applied to predict cognitive levels of questions based on Bloom's taxonomy and the resulting questions were made available to the instructor through the LMS interface. Future research should focus on the use of convolutional reinforcement learning to establish its effectiveness in question classification as well as perform comparison with various algorithms.

CHAPTER ONE: INTRODUCTION

1.0 Introduction

This chapter covers the background of the study, problem statement, scope and justification. Background of the study looks into the practice and scenario description involved in exam generation. Problem statement identifies the gap, and outline the purpose and objectives of the study. The scope identifies the environments that the study covers whereas justification states how this study affects learning through enhancing exam generation process.

1.1 Background of the Study

Over time, there has been a notable increase in the number of students joining tertiary institutions. To manage the growth, institutions have responded by creating flexible learning patterns like introducing e-learning and embracing technology in digitizing learning activities. Automation has been adopted to enhance efficiency and effectiveness at a reduced timeframe (Balfe, Sharples, & Wilson, 2015). A flexible learning pattern is established that calls for a flexible examination pattern. Therefore, examiners require to re-think on an approach to cater for this increasing need of flexible examination pattern (Ndirangu, Muuro, & Kihoro, 2021).

Manual examination setting has for the longest period been effective and widely used. Setting examination is a process that consist of questions development, moderation, typesetting, proof reading and printing. Changing learning patterns has reduced the effectiveness of this process as the need for examination increases. The activity requires skilled persons in a specific domain under examination to provide expertise. Time is another resource that is consumed during setting of examination.

There are many types of assessment or 'testing' to access student's learning curves. However, the written examination is the most common approach used by any higher education institution for students' assessment (Omar et al., 2012). Many studies have sought to automatically classify exam questions based on Bloom's taxonomy (Abduljabbar & Omar, 2015). Researchers have used NLP, machine learning techniques, and rule-based approach techniques in questions classification and exam generation. Examination questions require to be well analyzed to fulfill the requirements by different education levels such as a Bachelor's degree or Master's level (Mohammedid & Omar, 2020).

Cognitive level is thinking skill that form the complexity of a question. Questions are said to be simple or hard based on this metric. Bloom's taxonomy provides a method of classifying questions into difficulty levels that are knowledge, comprehension, application, analysis, synthesis and evaluation. A standard exam focus in striking a balance on complexity when evaluating a certain domain. Experts were used to assign weights to the questions as they are able to classify questions in their domain into their difficulty level based on the Bloom's Taxonomy (Omar et al., 2012).

Some studies have covered on questions classification using various artificial intelligence (AI) techniques specifically supervised machine learning algorithms. The recent studies have recommended use of deep learning techniques to determine cognitive levels. This question complexity can be useful input in the process of examination generation to ensure the standard is achieved and maintained (Ndirangu et al., 2021.).

Setting exams is the process of preparing questions for use in assessing the concept taught and all the processes of setting exams should be made internally i.e. questions development, moderation, vetting by the external examiner, printing, and proofreading. The processes consume a lot of time and may, at times, subject examination for leakages in case they are mishandled (Ogula, Muchoki, Dimba, and Machyo, 2006).

Content validity, scorer reliability, discrimination, and objectivity are the four principles identified by Johnson (2001) that constitute a standard examination. Content validity is representative coverage of the whole course. Scorer reliability provides that if the script is subjected to two different examiners, they should arrive at the same score i.e., there shouldn't be a significant statistical difference in score. There should be a way to differentiate achievers and weak students to avoid discrimination. Objectivity provides that the test be fair to all irrespective of age, gender, religion, or any other natural distinction. Examiners should ensure that test by students at the same level or class evaluates similar concepts to enforce objectivity with main focus on the cognitive levels in Bloom's Taxonomy (Johnson, 2001).

A quality exam should consider the six of Bloom's cognitive domains of knowledge namely; remembering, understanding, application, analysis, evaluation, and creation (Krathwohl, 2002). An exam comprises of questions and answers. Questions have different complexities (cognitive levels) and contain properties that include; mark(s)/weight and topic. The weight assigned to a question determines the score and is based on the level of study and question complexity as described by Armstrong, (2016) in the Revised Bloom's Taxonomy. Questions examine an area of study (topic) and their complexity indicates the difficulty level that is categorized as very simple, simple, moderate, hard, or very hard indicating that questions have different difficulty levels (Eldesoky, Aboutabl & Haggag, 2014). The difficulty levels build in increasing order from basic, rote memorization to higher (more difficult and sophisticated) levels of critical thinking skills. Pathan & Futane, (2019) determined that Blooms Taxonomy is a standard estimator for questions difficulty level during their creation. Therefore, cognitive levels must be clearly defined in the questions during examination generation to enhance standardization. However, if not properly considered, it may lead to an imbalanced test, i.e., containing many sophisticated questions making it hard for the students or vice versa.

Kurdi, Leo, Parsia, Sattler, & Al-Emari (2020) observed that questions construction is a complex process that requires training, experience and resources thus a challenge in developing examinations as questions and answers development need to be adequately addressed.

There are few experts in a particular domain who can develop a standard exam in many institutions especially private, as they are faced by financial constraints (Manogharan, Thivaharan, & Rahman, 2018). It is expensive and cumbersome to seek their service in examination generation. The experts can, however, be put to use to come up with standard questions which can be used in a system to generate standard exam. The process of setting exams manually adds in the lecturer's workload and consumes a significant amount of time which can be used for other activities when a system is used.

1.2 Problem Statement

Researchers have been involved in the automatic examination generation with the majority focusing on the multiple-choice and what, where, when, how ("wh") questions as demonstrated by Ali, Eassa, & Hamed (2018). Majority of the researchers focused on vocabulary assessment and understanding, with few concentrating on the complete spectrum of Bloom's Taxonomy. Most studies have covered Bloom's taxonomy to test questions' cognitive with some using machine learning algorithms (MLA) with focus on deep learning not fully exhausted.

Examination generation automation process has had tremendous progress but it can be improved to ensure production of a quality standard exam. A lot of studies have focused on multiple choice questions with less incorporating other question types (Ndirangu et al., 2021). Rahim et al., (2020) recommended the use of deep learning algorithms in questions classification. The research embraces the use of Artificial Intelligence (AI) by focusing on the use of Natural Language Processing (NLP) and Artificial Neural Network (ANN) in the process of questions classification using Bloom's Taxonomy for the purpose of automating generation of standard examination.

1.2.1 Purpose

This research proposes to automate generation of examination using NLP and ANN in line with the Bloom's taxonomy.

1.2.2 Objectives

The objectives of this study are:

1. To investigate different automation technologies used to auto-generate examinations.
2. To design a model which can auto-classify exam questions using Bloom's Taxonomy, ANN and NLP.
3. To develop a machine learning model for questions classification using the two algorithms.
4. To test the model, and integrate it with a Learning Management System (LMS).

1.3 Scope

The study has been limited on the use of Natural Language Processing and Artificial Neural Network development of a model that auto-classify questions using the Bloom's Taxonomy. Moodle learning management system(LMS) has been used to integrate the model developed for questions classification.

1.4 Justification

There is a growing number of students joining institutions of higher learning. Institutions have responded to the rise by providing flexible learning patterns including eLearning. As a result, students have been allowed to pursue different number of courses, finish and sit for exams at different timeframes. The different modes of study and existence of few experts in some domains has raised the need to have examination process automated to save on time and cost used in generating exam as well as ensure quality standard exam.

Therefore, there is need to generate quality standard exams using machine learning techniques as it will help examiners to administer a standardized exam that is objective, non-discriminatory, scorer reliable, and achieve content validity with less people's involvement.

1.5 Significance of the Study / (Rationale)

This study directly impacts the:

1. Examination coordinators – Having a system fed with proper questions and ready to generate quality exams on need basis has been the dream of any examination coordinator. This study helps exam coordinators to produce tests with questions cognitive levels factored at a click of a button reducing the timeframe and people's involvement.
2. Lecturers – Every time a lecture facilitated a unit/course, it is expected that an exam was prepared for that. Having the system in place saves time and ensures that lecturers stick to their primary objective which was to facilitate learning.
3. Student – Having examination ready for administration on need basis provides students autonomy to study and be examined.

The study indirectly affects the institution administration by saving them the stress created by the process of setting, proofreading and printing exams that include ensuring exam leakage had lowered chance of occurring.

CHAPTER TWO: LITERATURE REVIEW

2.0 Introduction

This chapter discusses the related literature on the automation of examinations questions classification based on Bloom's Taxonomy. First we discuss different ML algorithms in relation to classification of exam questions. Secondly, we discuss different exam questions classification models and thirdly a theoretical framework.

2.1 Machine Learning Overview

Machine Learning (ML) is described as a type of Artificial Intelligence (AI) that provides a computer with the ability for being trained without being programmed (Praveena & Jaiganesh, 2017). ML algorithms are broadly classified into supervised learning, unsupervised learning and reinforcement learning. Machine learning identifies the patterns that exists in the data and model them to enable prediction of new-fangled data.

Technological advancement has led to increased use of artificial intelligence on the classification of examination questions (Acemoglu & Restrepo, 2019). The application of AI in automation led to a reduced business operational and vocational cost (Shekhar, 2019). AI has led to increased accuracy and efficiency in as much as use of machine learning abilities are not fully exploited. Institutions are now focusing on fully utilizing the abilities in the technologies to bring out the best in the examination questions classification.

Supervised learning is described as a technique that involve construction of predictive models by identifying patterns from a great number of training example in which every example has a ground-truth inform of a label (Zhou, 2018). Each example contains an input object, usually a vector quantity, and an output value (Praveena & Jaiganesh, 2017). A supervised learning algorithm does analyze the training data and formulate a contingent function for new examples mapping.

Unsupervised ML provides analysis of raw datasets to generate analytical insights from unlabeled data (Usama et al., 2019). Reinforcement Learning (RL) is described as one where an agent interacts with an unknown environment in an effort to figure out the most probable correct answer and receives feedback in the form of a reward or punishment from the environment (Naeem et al., 2020). The feedback was used to train and collect experience and knowledge about the environment.

2.2 Machine Learning in Examination Generation Process

Machine Learning techniques was adopted in a number of projects in the questions classification, test generation and even in other aspect of online tests. This section covers different ways in which ML has been implemented by various researchers in an examination generation process. It comprises ML in e-learning, ML in questions generation and ML in examination generation.

2.2.1 Machine Learning in E-learning

E-learning is described as the use of electronic applications and processes to learn that also include the use of IOT (Internet Of Things) (Soni, 2019). Applications and processes used in e-learning are web-based learning, computer-based learning, virtual classrooms, and digital collaboration. Delivery of content is done digitally through the internet, intranet, extranet, satellite TV, and CD-ROM with multimedia capabilities (ISP, 2004). E-Learning students' studies at different paces, therefore, register units based on the capacity one could handle, making them depict some characteristics like handled differing number of units, and completed at different times thereby bringing outstanding traits to students under this mode of study.

E-Learning is described as a flexible model of facilitation and examination which should accommodate the prevailing needs like impromptu examinations that do not lose the focus of determining the success of learning (Meskhi et al., 2019). Ordinarily, students are tested using a CAT, assignments, and final examination. The process is uniform across many institutions with the adoption of AI in e-Learning to facilitate efficient learning and at convenience. E-Learning examiners should be ready to administer CATs and Examinations for students on a need basis as students study at different paces using personalized learning approach (Tang et al., 2019). The properties of a good exam should be maintained to provide high quality standards of examination. The existing process has made the burden of the examiner more and sometimes unbearable. The whole process of setting, storing, and retrieving an exam should be accomplished by embracing technology (Rao, Sai, Sandeep, & Jayadeep, 2020).

2.2.2 Machine Learning in Questions Classification

Questions classification based on Bloom's taxonomy has received considerable critical attention in recent years where researchers have used different techniques and features (Mohammed & Omar, 2020). In their study, K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Logistic Regression (LR) machine learning algorithms and natural language processing have been used. The study combined two features that is word2vec and TFPOS-IDF (W2VTFPOS-IDF) in questions generation.

Verbs and actions were used to demonstrate different levels of learning. The solution was based on the classification of the action verb of the questions or learning outcome statement (LOS), to classify the whole question or LOS into a more accurate level (Diab & Sartawi, 2017). Action verb classification algorithm was applied on the verb lists from questions and LOS to compute the maximum similarity for every level of the cognitive domain and used a rules-based approach. The finding concluded that the approach can be used to provide more accurate verbs and, in turn, provide more accurate intended mental skills.

A document analysis method indicated that teachers were asking many questions at the first three levels of Bloom's Taxonomy. Most teachers fear that the student may not pass the test and therefore, resolve to set questions on the **low** cognitive levels. There is a need to consider the questions at the higher cognitive levels to facilitate critical thinking (Karamustafaoglu, Karamustafaoglu, Bacanak, & Degirmenci, 2011). According to Buick (2011), surface learning is entertained by assessment strategies that reward low-level outcomes.

A comparative study of SVM and K-NN to achieve better performance and high quality has been done where grammar checks and in context check were used. The classification was used to test the student level, and skills gained compared to Bloom's taxonomy cognitive levels (Patil & Shreyas, 2018).

Support Vector Machines (SVM) algorithm has been used where the classification algorithm was divided into three steps; text representation, SVM classifiers construction, and SVM classifiers evaluation. This technique was evaluated by varying the frequency of stop words. The research observed that an increase in the number of words used to represent the question lowers SVM's performance. In conclusion, the number of stop words should be more than one for excellent performance and that reducing the number of stop words does not significantly improve performance (Yahya, Toukal, & Osman, 2012).

2.2.2 Machine Learning in Exam Generation

Computer technology has rapidly changed leading to the development of new ideas and algorithms including simulating the process of generating exams. Exam generation system need to accommodate MLA that can be used to generate exams needed and consider the difficulty level, topic, weight, and cognitive level of a question to examine learning successfully. The system need should be designed to put into consideration the approach, tools, and algorithms that are used in the development phase to enhance highest standard as the quality of E-Systems is determined by views and usages (Nabil, Mosad, & Hefny, 2011).

Filtering criteria was adopted that: exclude or include past semester questions; the total number of items per paper; item complexity; maximum items per topic; paper topic settings; test paper generation; and items analysis with s question bank used for questions storage (Yusof, Lim, Png, Khatab, & Singh, 2017). An automated paper generation system that focuses on controlled access, question randomization, and user roles has been done by (Bhirangi & Bhoir, 2016). Focus on the cognitive level (difficulty level) was not considered. The software has been developed using Java programming language and MySQL database. The algorithm used is improved randomization of questions.

Artificial intelligence, randomization, and backtracking algorithms have been used in a project to automate the exam generation process by (Cen et al., 2010). Technologies used in the system are MVC pattern in JSP view, JavaBean models, the Servlet Controller, MySQL as the database, CSS + DIV for layout, and JavaScript for support of page details. The study did not focus on the cognitive level (difficulty level) of questions.

Random selection and backtracking algorithms have been used to determine the cognitive level of questions. The weight of the questions was computed as a percentage and CSV (Comma Delimited

Values) used to input the structure of the paper. The process can be automated to enhance storage and intelligence that would lead to the locking of questions recently used for examination. The research concluded by recommending future systems to consider using Natural Language Processing (NLP) (Joshi, Joshi, & Doiphode, 2016).

Package exams were used to provide software infrastructure for scalable exams, associated self-study materials, and joint development. The software used maintenance, variation, and correction as design principles. Technologies used are LaTeX, and R where questions were separated into answers and solutions section and meta information collected. Question and solution description were encapsulated in LaTeX and every exercise contained in a separate sweave file; therefore, the need for separate files. This method was used to make a custom application for statistical processing exams. Artificial intelligence (MLAs), Cognitive level, weight, and difficulty level were not explicitly considered (Grun & Zeileis, 2009).

Natural language processing has been used used to process text while Named Entity Recognizer and Semantic Role Labeler used to identify the semantic relation in a system. The main focus was to generate simple true or false questions or those that require a one-word answer (Rakangor & Ghodasara, 2015).

Hameed & Abdullatif (2017), developed an online system that utilized web-based technologies; PHP, MySQL database. Three types of questions were generated that were; true/false, multiple choices, and image matching. However, artificial intelligence, and cognitive level were not used in the system.

A rule-based classification approach was used to classify exams with an established model that enabled adjustment of the paper quantitatively. Though the model worked to classify the questions using cognitive levels, the research concluded by recommending the use of machine learning techniques to increase performance (Kumara et al., 2019).

2.3 Natural Language Processing in Exam Generation

Natural Language Processing, is a description of human languages in which the questions find context. It is a theoretical motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistics for the purpose of achieving human-like language processing for a range of tasks or applications (Liddy, 2001). The major goal for NLP is *to accomplish human-like language processing*. It is done into two steps language processing and language generation. Computers need to understand and conceptualize the questions to generate the required model for the machine learning algorithm. Therefore, NLP helps to read, decipher, understand, and make sense of human languages in a manner that is valuable and understandable by computers (Hirschberg & Manning, 2015). NLP is used as a preprocessing tool to clean the human-readable text and convert it to a more usable text by removing stop words, duplicated and irrelevant, and noisy data for instance Analysis is converted to a vector [1. 0. 0. 0. 0. 0.].

2.4 Artificial Neural Network (ANN) Algorithm

ANN, is a useful model in problem-solving and machine learning and works in manner to manage information similar to biological nervous systems function of human brain (Abiodun, Jantan, Omolara, Dada, Mohamed, & Arshad, 2018). The concept behind the algorithm is taken from biological neural networks (Gupta, 2013). It comprises of a large collection of units interconnected using a certain pattern to allow communication. The units operate in parallel and are also known as nodes or neurons. A link is used to connect nodes associated with a specific weight that inhibits or excites the signal being communicated. Every neuron has an activation signal, output signal, and an activation rule.

The architecture consists of an input layer, a hidden layer (can be more than one), and an output layer. ANN layers are independent of one another with each hidden layer used to learn patterns from the input layer and pass on to the succeeding layer to use it as its input. The activation function is used to capture the non-linear relationship between the inputs and convert it into a more useful output. When ANN is used to classify, input node match the features as the output nodes match the output classes (Abiodun et al., 2018).

ANN layers are organized in a manner that each unit in a layer relates to all other units in the layers which is termed as feedforward neural network (FFNN) (Abiodun et al., 2018). The connection to layers may be different based on the weights of the network connection. Information in the FFNN was transmitted in only a single direction, which is from the input layer to hidden layer(s) and to output layer. The other organization of the ANN units termed as back propagation neural network (FBNN) produces a coordinated graph in sequence that provides feedback that demonstrate dynamic terrestrial behavior for a time sequence (Abiodun et al., 2018).

Deep learning (DL) consist of complex multi-layers of artificial neural network (ANN) with more neurons to express complex models that needs more computing power to train and has an automatic feature extraction (Abiodun et al., 2018). Future researchers should explore deep reinforcement learning in combination with other algorithm or model, or neural networks to improve the system (Rahim et al., 2020).

2.5 Theoretical Framework

Machine learning involved training a machine to search for patterns in data and improve from experience and interaction. The dataset is selected and grouped into training, validation, and testing sets. This data is passed through NLP for preprocessing, and finally, neural network algorithm used to generate a model for question classifications which was used during examination generation (Immanuel, 2015).

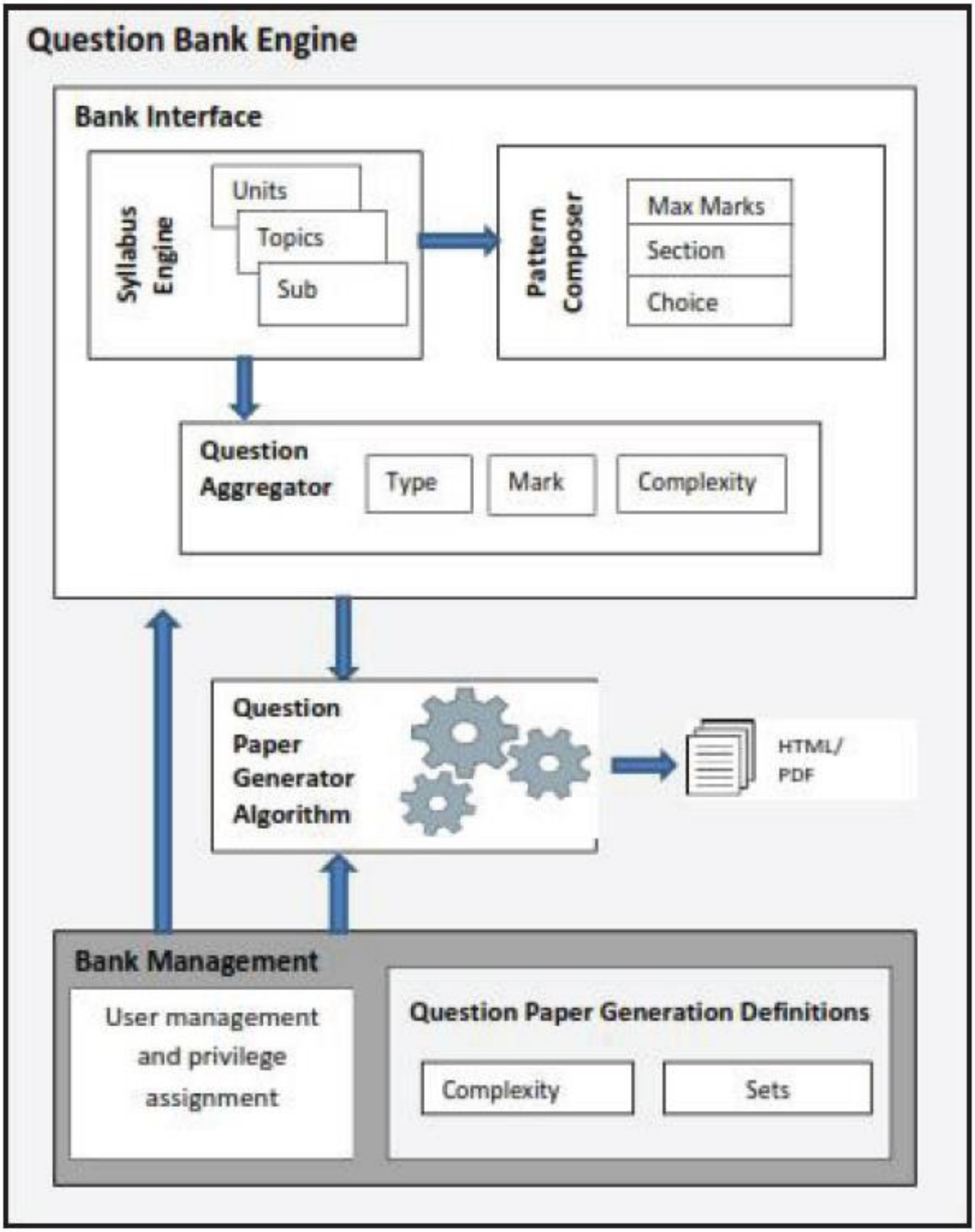

Figure 2.1: Theoretical Model (Immanuel, 2015)

CHAPTER THREE: METHODOLOGY

3.0 Introduction

The research adopted the iterative methodology of software development as all requirements were not adequately defined. The chapter discuss the different stages involved in iterative methodology namely; system architecture, data preparation, requirement specifications, architectural design, detailed design, coding & testing, integration & testing, and operation and maintenance as illustrated in figure 3.1.

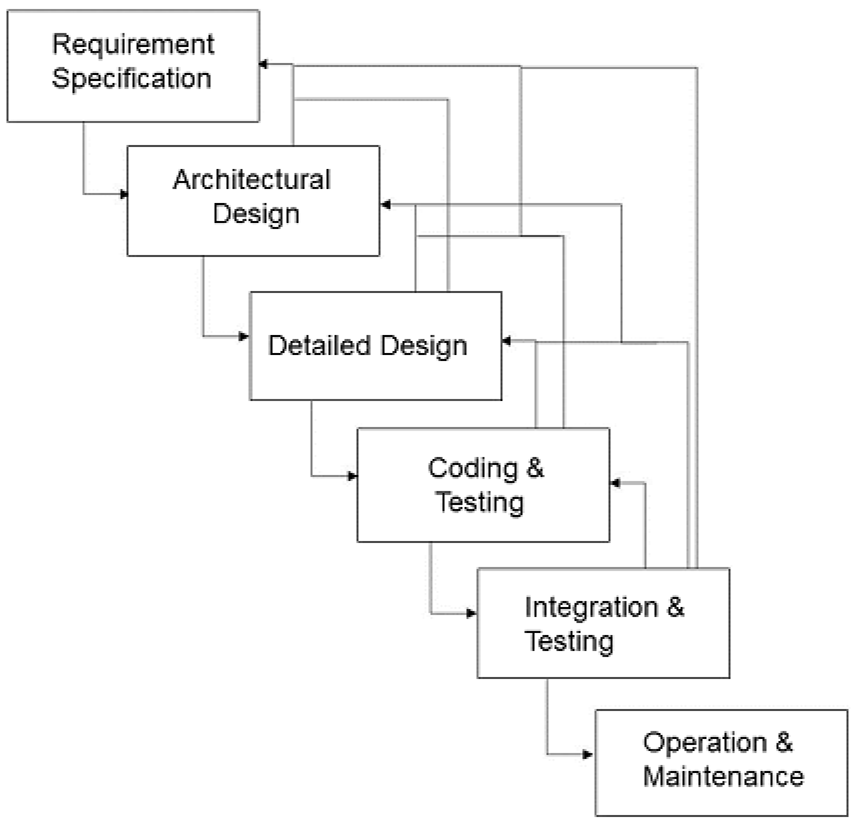

Figure 3. 1: Iterative Method

3.1 Requirements Specification

The system information was gathered and analyzed to develop user and technical requirements. The e-learning domain was adopted for integration with to test the effectiveness of our model in E-Systems. Training data comprising classified questions was sourced from online source that is

Google's datasets search engine for use during model development. The procedure of setting exam was gathered using unstructured interviews and documented using a flow chart as illustrated in figure 3.2. This data exists in word format and therefore needed to be converted to a CSV file for consumption by our model.

3.2 Architectural Design

The system was implemented by segmenting it into 3 phases. During the first phase, the model was developed through data collection, coding, training and testing the model using python programing language and Google colaboratory. Phase two involved developing an API that takes the model and consumes a question to provide the question's cognitive level using python rest framework, and the question classification model. The last phase was to incorporate the system with an online LMS using plugin developed by use of PHP programming language and Moodle LMS. The exam generation system consists of a question bank, composer, aggregator, test generation among others.

3.2.1 Question Bank

Successfully created questions were saved in a database as records regarding the question and their activities. A predefined way was used to save questions to enable retrieval and ensure uniformity. Question bank is thus a collection of questions from various units and topics into a single platform as shown in figure 4.2.1.

3.2.2 System Architecture

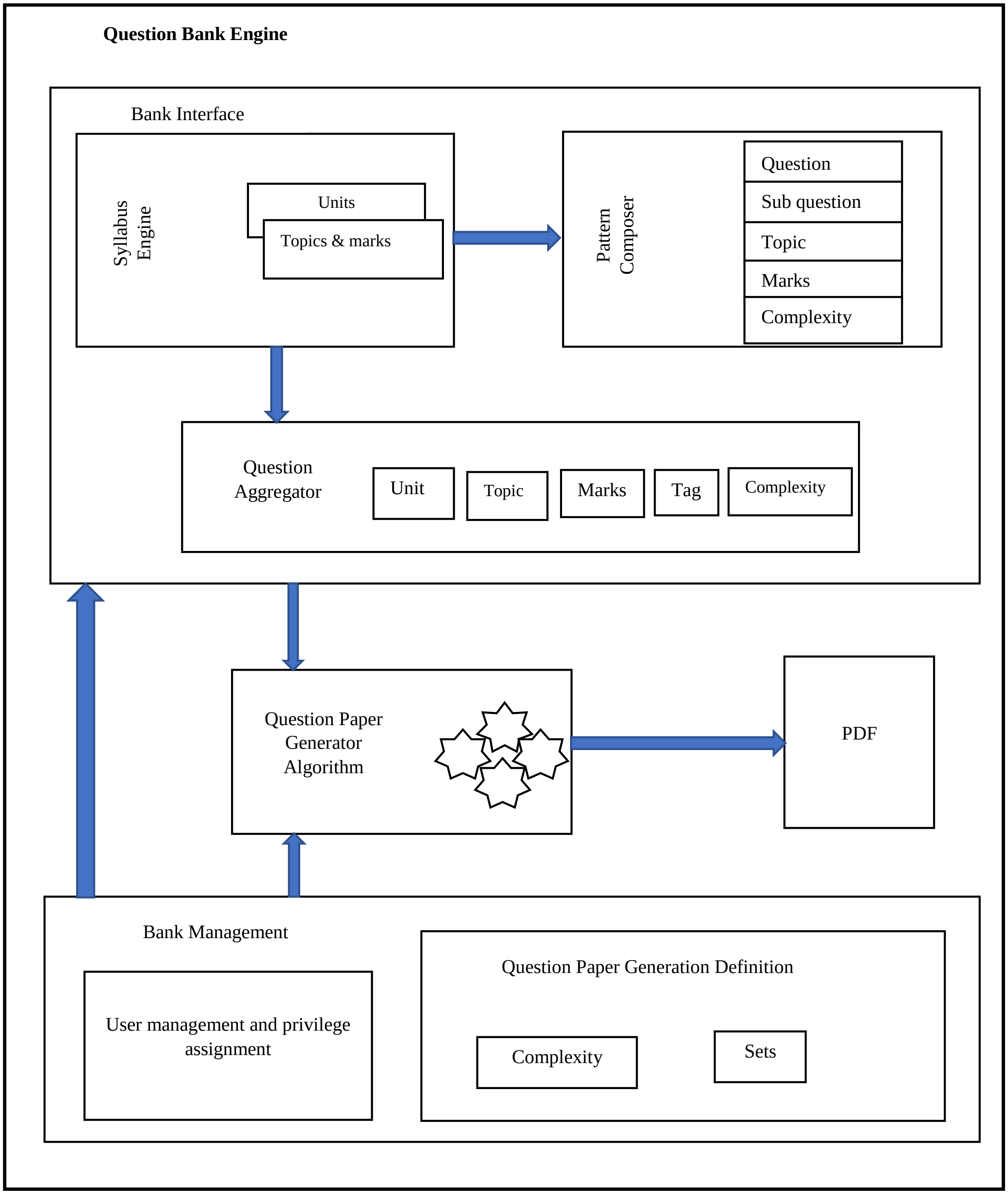

Figure 3. 2: System Architecture

3.3 Data Preparation.

Relevant data was gathered from Google's datasets search engine and checked for conformity with the requirements that is properly labelled questions into Bloom's Taxonomy classes. Questions that had cognitive level defined using Bloom's Taxonomy were selected and grouped into training, testing, and validation sets by splitting it into 75% training and 25% validation. This grouping was a ratio of the dataset where the ratio was varied until optimum solution for the model was established.

3.4 Detailed Design

The software design entailed the design of the database, user interface output and dialogue designs was defined to meet the necessity for the design. Database was represented using an ER-diagram to show the structure and relationships. A design for model was developed using the artificial neural network architecture. User interface (UI) designs was developed for use during system development represented by data flow diagrams and use case diagrams as shown in figure 4.5 and 4.4 respectively.

3.5 Coding and Testing

A database of questions was developed based on the database design as shown in figure 4.1. A relational database (MySQL) was used to store information about questions and answers. Coding of a model was done with respect to the model design using neural network algorithm, and NLP. Data was subjected to NLP to decipher the human readable text and convert it into information that a machine can understand. The resulting data was subjected to neural network algorithm and parameters varied to come up with a model that provides optimal results. The algorithm underwent training to make it learn and generate a model for question classification. An application programming interface was developed for the model to allow questions classification using a defined user request. Finally, a Moodle plugin for question classification was developed and integrated with Moodle LMS.

To ultimately achieve the objectives, the following additional tools and technologies were applied.

1. Google Colaboratory – Used for model development and testing. It offers ability to develop machine learning model in an online platform that is accessible at any place.
2. Python Programming Language – Python has well developed tools that allows the use of NLP and Neural Network algorithms effectively and efficiently.
3. Django-rest framework – The technology was used in the API development.
4. PHP – Integration to Live system. This technology was useful in the integration process with a PHP developed exam generator system.
5. Database / Questions bank – For questions storage. The system stored question's metadata and the question text in a relation database.

During model development, a code that takes classified questions as input and pre-process it to data that can be understood by machine was developed. The code also comprised neural network section that take pre-processed data as input and use the set criteria to produce a model that was able to classify other sets of questions within the domain. The criteria consisted of an activation function and hidden layers that were varied until an optimal performance of the model was established.

3.6 Integration and Testing

Validation data obtained from splitting the training data was used to determine model precision and accuracy. Unit testing was done on during model development to ensure every module perform as per the expectation and evaluate its suitability for use in questions classification. Integration testing was done to ensure that modules that share information communicate effectively. System testing was performed to make sure that the whole system conforms to the user requirements. Testing data was passed on the generated model to determine its effectiveness and precision.

Precision is the ratio of true positives out of the sum total of true positives and false positive see Equation (1).

$$Precision = \frac{True\ positives}{True\ positive + False\ positive} \tag{1}$$

Precision indicates how accurate prediction was, in our case it was at 71%. Recall is a parameter used to measure the ratio of correctly predicted outcomes. It indicates the specificity or sensitivity see Equation (2).

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{2}$$

Accuracy is the ratio of correct predictions out of all predictions made by an algorithm. It can be calculated by dividing precision by recall or as 1 minus false negative rate (FNR) divided by false positive rate (FPR) see Equation (3).

$$Accuracy = \frac{(True\ positive + \ \ True\ negative)}{True\ positive + False\ positive + True\ negative + False\ negative} \tag{3}$$

The F1-score combines these three metrics into one single metric that ranges from 0 to 1 and it takes into account both Precision and Recall see Equation (4).

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

CHAPTER FOUR: RESEARCH FINDINGS AND DISCUSSION

4.0 Introduction

This chapter discusses how the system design was implemented, tested and also discusses the results obtained after testing the system. System design involve coming up with system components, modules, interfaces, and data for system to actualize specifications. The designs set the baseline as part of the inputs when translating them to a working program by coding. This consisted translating the human language to a machine understandable format to enable computers to process instructions provided. The developed software was subjected to levels of testing to check its performance and effectiveness.

First we discuss the design specifications such as requirement specifications, secondly the system design in terms of database, interface, thirdly we discuss the ML design implementations and finally the results of testing the model in the database of questions.

4.1 Requirement Specifications

All requirements identified during design and analysis of the system were collected and categorized into user and technical requirements. Specifications were drawn from the requirements.

4.1.1 User Requirements

These were the functional requirements as the user envisioned the system. They express the processes and their behavior in terms of condition, throughput etc. The user requirements identified included the following;

1. Questions with labels on classification to be used for model training.
2. The model should predict new questions cognitive levels
3. The system should provide a way to create questions on an interface.
4. The system should enable update for questions already created.

5. The system should enable management of the questions
6. The system should enable lecturers to define key unit information such as topic, marks etc.
7. The system should allow lecture to define exam instructions that fall under preamble.
8. Lecturers to be enabled to come up and manage exam structure led by questions cognitive levels.
9. The system should be accessed through the web browser.
10. The students should be able to attempt the test by a click of a button.
11. The system should factor in parameters like topic coverage, cognitive levels of questions, marks among others for testing.
12. The system should ensure that only authorized users are allowed.
13. Users should be able to manage their profile information.

4.1.2 Technical Requirements

These are the non-functional requirements that the system fulfil. They are the conditions for proper operation of the system. The following list outlines the technical requirements which were identified;

1. The model should be served by an API for access by client app for questions classification.
2. The system should be accessed via a web browser
3. The system should be able to run on a cross-platform environment.
4. The system should be able to have high level of security to keep information safe.
5. The system should be locally or online hosted
6. The system should be able to run on Linux, Windows and Mac OS X
7. The system should have an authentication feature.
8. The system should ensure role based access.

4.2 System Design

This section provides all system designs including for databases, artificial neural network model as well as user interface designs. The system was built based on the architecture shown in figure 3.1 that forms core of the designs. The designs enabled translation on the system during implementation.

4.2.1 Neural Network Model

The design in figure 4.3 was used in the development of questions classification model using artificial neural network and natural language processing. It indicated the input, neurons, hidden, activation function and output layers used in development of questions classification model.

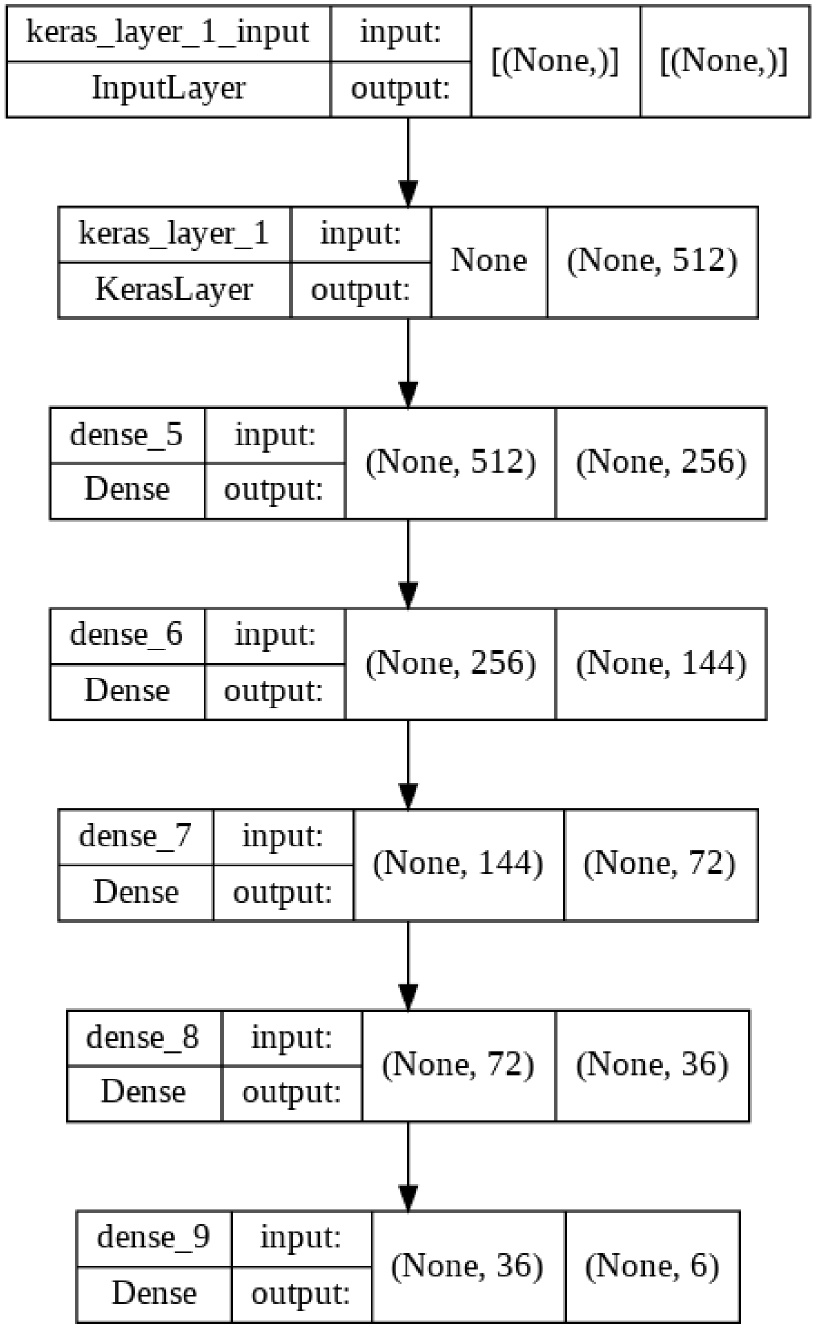

Figure 4. 1: Neural Network Model

4.2.2 Use Case Diagram

The user interface diagram was designed to indicate interaction of the lecturer with the examination system from questions creation to generation of a standard test.

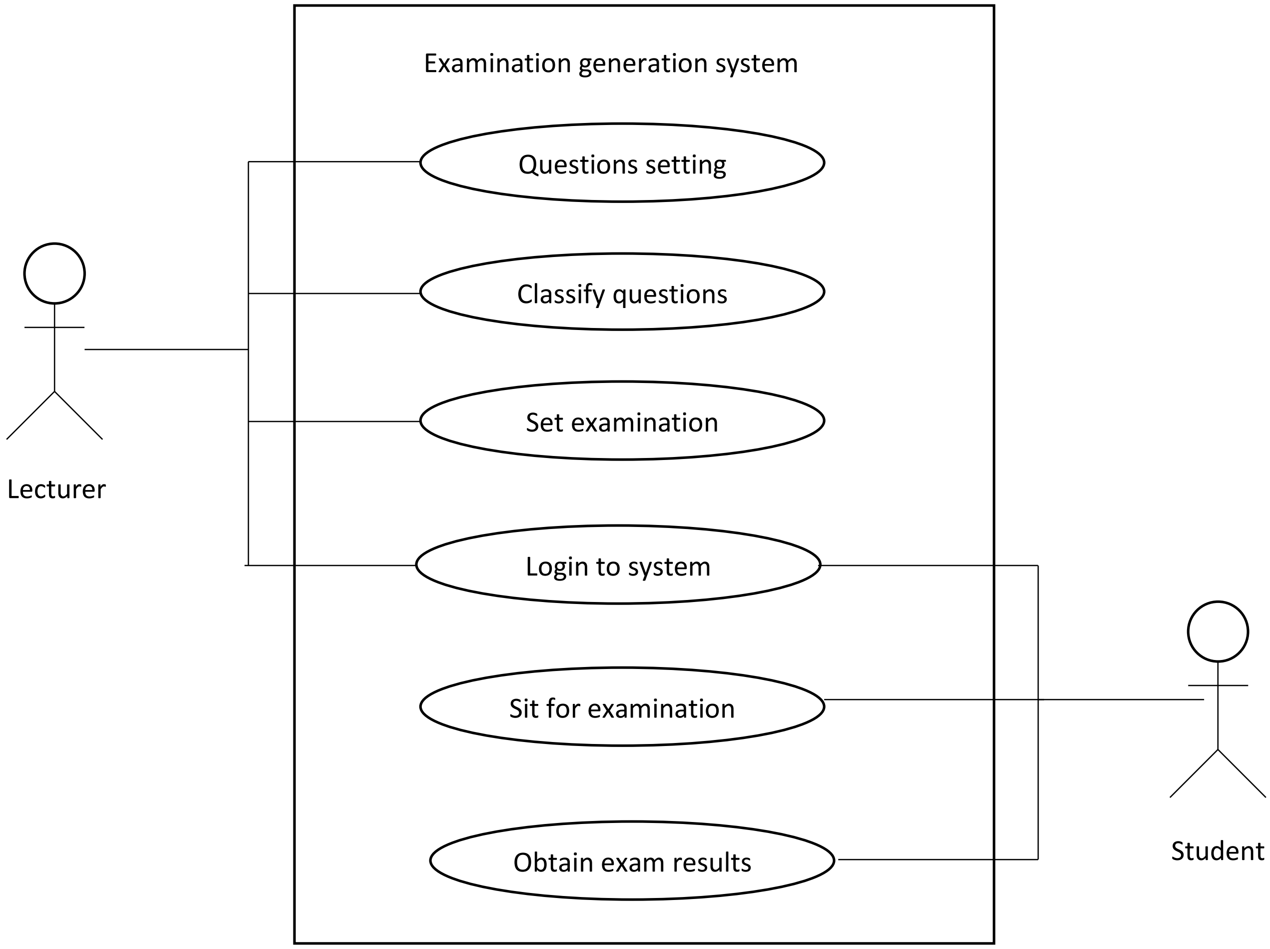

Figure 4. 2: Use Case Diagram

4.2.3 Data Flow Diagram

This user interface design was a visualization of the flow of information within the system and how it changes, stored and leave.

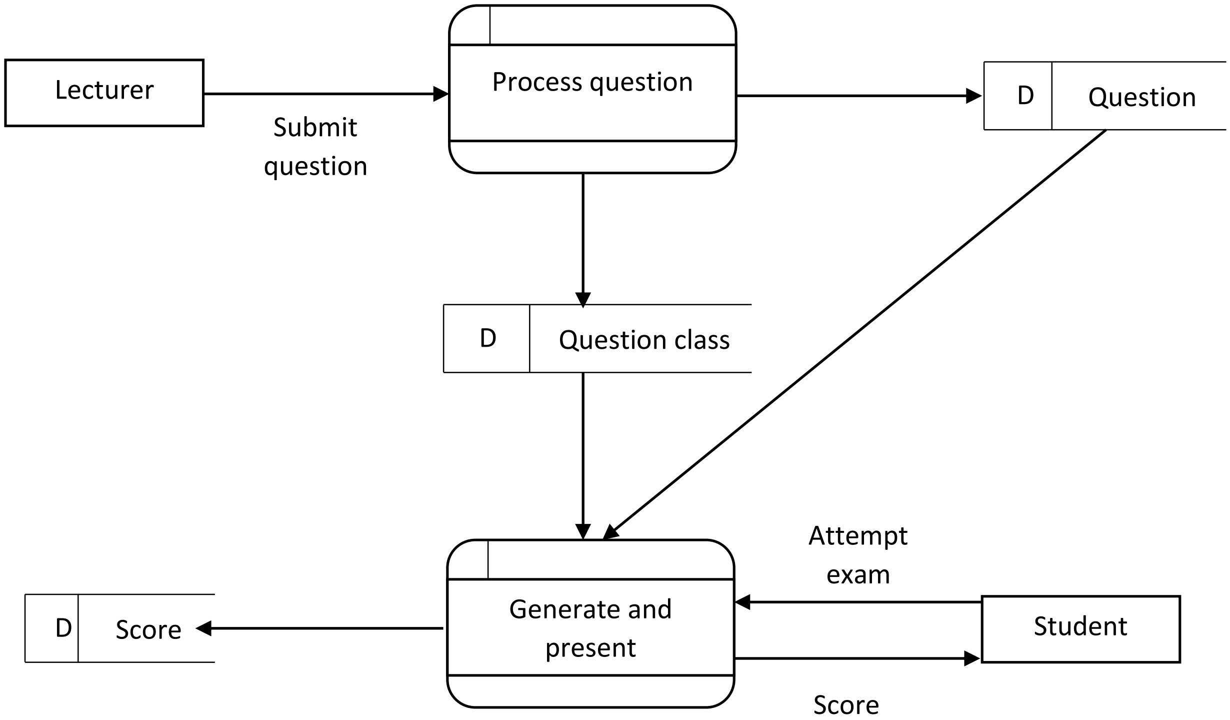

Figure 4. 3: Data Flow Diagram

4.2.4 Database Design

The database was designed to cater for files and raw data required to be captured by the system. This means that the database should capture the question, answer and the metadata for the question. Two designs for database were done namely; database of files and relational database.

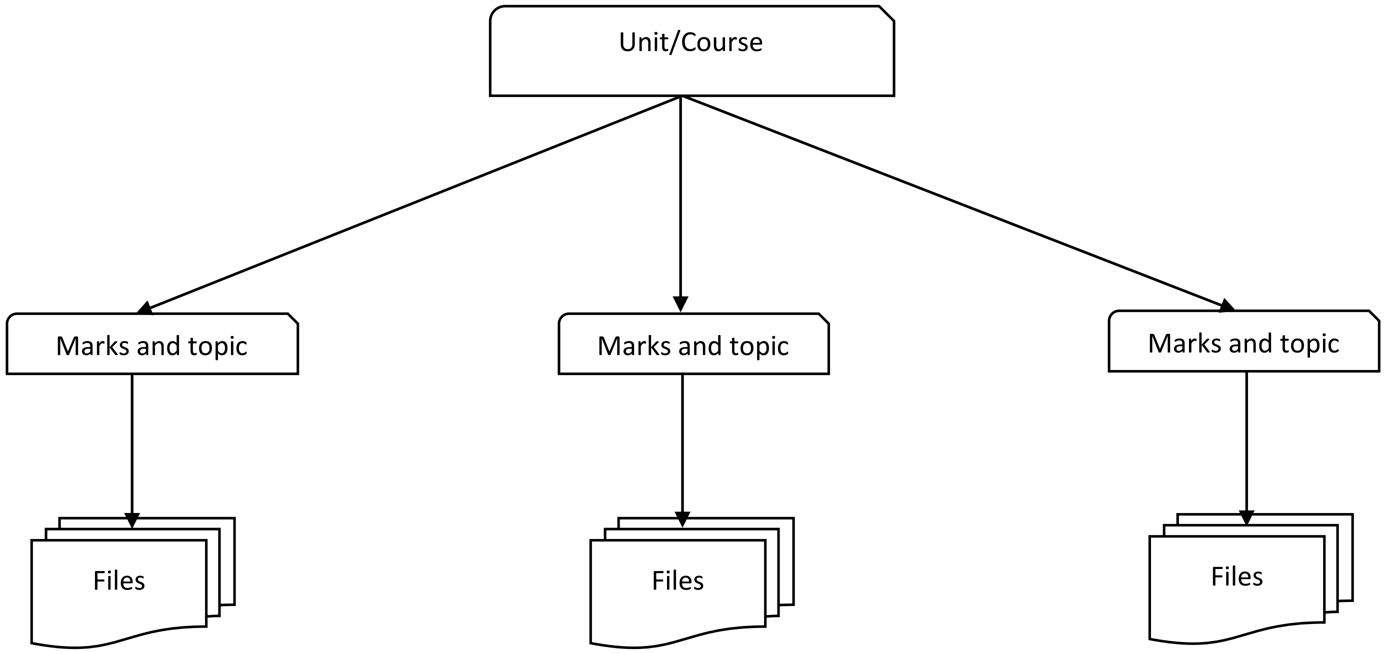

Figure 4. 4: Database of Files Design

The database of files was used to store the question details that include question text and answer(s) part. The relational database store questions meta data that is information about the questions such as topic, cognitive level, marks etc. This helps to keep track of the changes and area the question is stored.

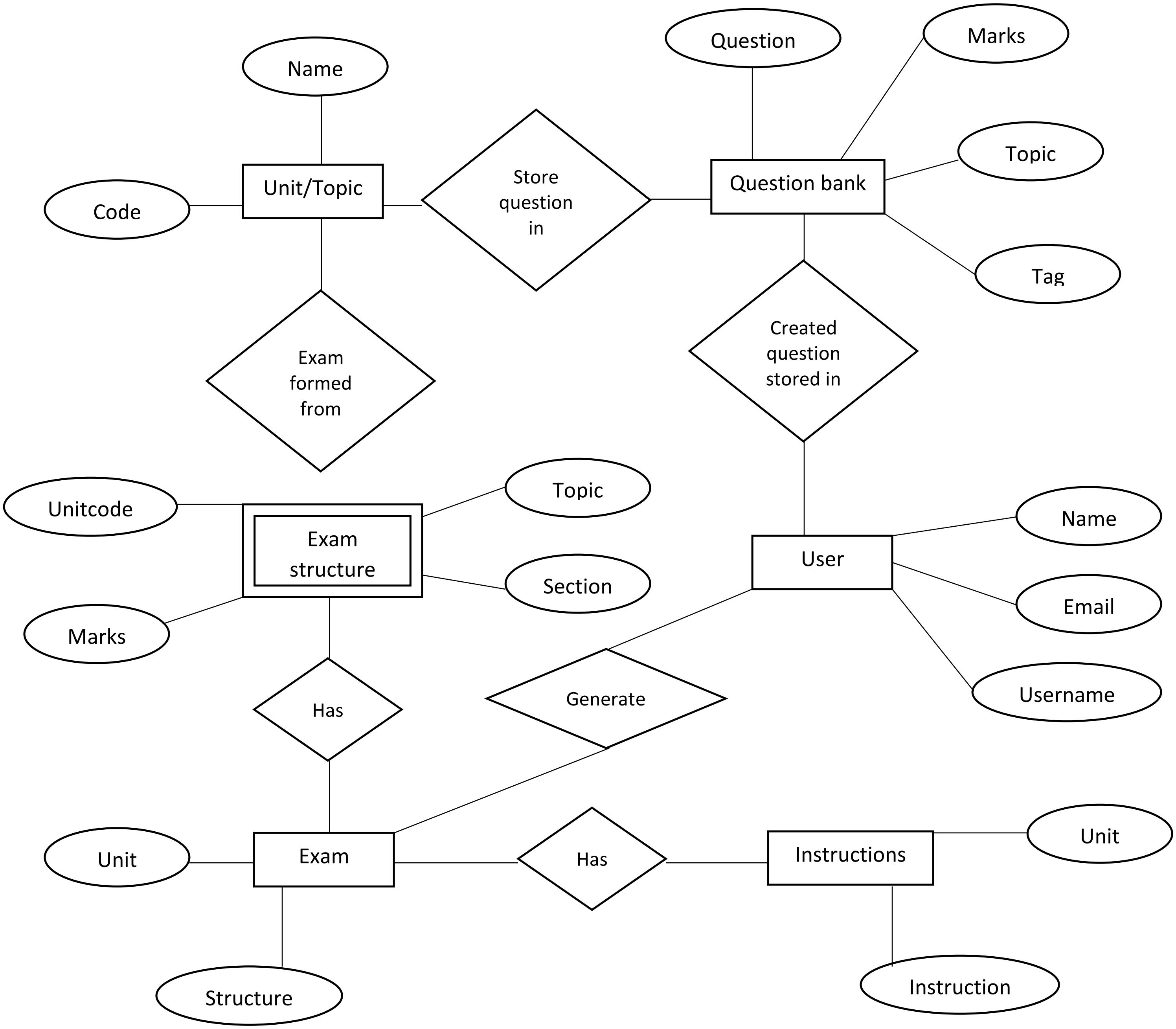

Figure 4. 5: ER - Diagram

4.3 System Coding

This comprised the process of translating the design and converting to a functional program. A suitable programming language was adopted and project converted to small programs that were easier to code. The section covers the process of building question classification system and embedding in an exam generation system.

4.3.1 Questions Classification Model

The model was developed by use of python programming language, libraries and Google collaboratory platform. It involved data gathering and organization, preprocessing, actual model development and model testing.

Data Gathering and Organization

Relevant data for questions classification was obtained from Google's datasets search engine and proper alignment established for easier processing of information. The data was structured as shown below, grouped into training and testing sets and stored in csv files ready for processing.

Table 4. 1: Sample Questions for Training Set

Sno	Class	Question
1	Knowledge	About what proportion of the population of the US is living on farms?
1	Comprehension	Compare Calliope with Howie. Use the word bank.
1	Application	Apply laws of statistics to evaluate the reliability of a written test.
1	Analysis	Analyze safe and dangerous aspects of these features.
1	Synthesis	Apply and integrate several different strategies to solve a mathematical problem (not according to one formula(
1	Evaluation	After designing an experiment, examining the results, and drawing conclusions, determines in what ways the experiment could be conducted more effectively in order to draw more productive conclusions in the future.
2	Knowledge	Correctly label the brain lobes indicated on the diagram below
2	Comprehension	Compare historical events to contemporary situations
2	Application	Apply shading to produce depth in drawing.
2	Analysis	Analyze the characteristics of frogs.
2	Synthesis	Can you create new and unusual uses for . . .?
2	Evaluation	After examining the videotape of a play in a football game, determine the degree to which the defensive team performed effectively and
3	Knowledge	Define compound interest.

3	Comprehension	Complete an analogy (analogy tasks are inference tasks)

Table 4. 2: Sample Questions for Testing Set

Sno	**Class**	**Question**
1	Knowledge	List two reference parameters in the setHour function
1	Comprehension	Can you explain what is happening . . . what is meant . . .?
1	Application	Design or sketch a marketing strategy for your product using a known strategy as a model.
1	Analysis	Analyze the selected information.
1	Synthesis	Create several different strategies to solve a mathematical problem.
1	Evaluation	Can you defend your position about ... ?
2	Knowledge	Explain briefly the meaning of the following terms: Demography, Consumer sales promotion , and Media mix
2	Comprehension	Class Facility is an abstract class. Explain what this means
2	Application	Sketch an experiment to see how plants grow in different kinds of soil.
2	Analysis	Can you discriminate the difference parts . . . ?
2	Synthesis	Can you propose an alternative plan to . . . ?
2	Evaluation	Can you assess the value or importance of . . . ?
3	Knowledge	Label the parts of the diagram

Pre-processing

Collected and organized data was passed through a pre-processing phase to:

1. Clean data by removing Unicode’s, symbols among others that machine translation may not be able to process.
2. Convert human readable text to machine understandable language. For instance, based on figure 4.6 questions labels were converted as follows:
    - Analysis converted to [1. 0. 0. 0. 0. 0.] vector;
    - Comprehension converted to [0. 0. 1. 0. 0. 0.] vector;
    - Application converted to [0. 1. 0. 0. 0. 0.] vector etc.

```
['Analysis', 'Comprehension', 'Application', 'Comprehension', 'Knowledge', 'Application', 'Knowledge', 'Analysis', 'Knowledge', 'Knowledge']
Text to number
[0 2 1 2 4 1 4 0 4 4]
Number to category
[[1. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0.]
 [0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0.]
 [1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1. 0.]]
```

Figure 4. 6: Conversion of Text to Figures that a Machine Understand

Actual Model Development

This section we discuss the processes that occurred during development of the model together with the technologies used. The initial step was to load two datasets and check its conformity with the expected format by checking its metadata like total number of questions per category which resulted to 600 questions for training and 141 for testing. The questions were labelled into the six classes as shown below.

```
===========Train Data =========
Comprehension    100
Evaluation       100
Analysis         100
Synthesis        100
Application      100
Knowledge        100
Name: Class, dtype: int64
600
==============================
===========Test Data =========
Synthesis        30
Knowledge        26
Evaluation       24
Analysis         23
Comprehension    23
Application      15
Name: Class, dtype: int64
141
==============================
```

Figure 4. 7: Total Number of Questions per Category and Class

The training data was then separated into 75% training and 25% validation sets separated as shown in fig 4.8.

```
Train data len:450
Class distribution: Counter({'Analysis': 75, 'Comprehension': 75, 'Application': 75, 'Knowledge': 75, 'Evaluation': 75, 'Synthesis': 75})
Valid data len:150
Class distribution: Counter({'Synthesis': 25, 'Knowledge': 25, 'Evaluation': 25, 'Comprehension': 25, 'Application': 25, 'Analysis': 25})
```

Figure 4. 8: Separating Training and Validation Sets

Model Training

This stage entailed defining and actualizing the model by coming up with the necessary code and implementing it to realize the model that classify a new question into one of the the Bloom's taxonomy class.

The input was a labelled question into one of the Bloom's taxonomy class, that is in the human readable format. Cleaned question text was converted into embedding vectors by use of a pre-trained text embedding as the first layer of our neuro network. Here we make use of a TesnsorFlow hub called google/universal-sentence-encoder/4 that takes care of further pre-processing as was trained on variety of data sources and tasks to handle variety of natural language tasks. It receives English text as an input and give a high-dimensional vector. A sample clean question and the resultant vector is shown below:

Clean Question: *['break down components standard film camera explain they interact make machine work']*

Resultant Vector:

```
<tf.Tensor: shape=(1, 512), dtype=float32, numpy=
array([[-4.90076579e-02, -6.55244291e-03, -5.82724288e-02,
        -1.07234316e-02,  3.37687805e-02,  2.82270052e-02,
        -5.63627891e-02,  4.68803607e-02, -7.97999054e-02,
         3.28014083e-02, -2.72867829e-02, -1.18662445e-02,
        -5.25323562e-02, -7.21590146e-02, -5.55040762e-02,
```

-3.30276564e-02, 1.35936560e-02, 6.86324853e-03,
 1.60365757e-02, 6.73190039e-03, -1.02198003e-02,
-6.75003603e-02, 5.07877171e-02, -2.52075829e-02,
 4.90289144e-02, -2.52602156e-02, -6.96410909e-02,
 6.25321716e-02, 9.23980493e-03, 7.00944662e-02,
-6.24308502e-03, -5.81241138e-02, -7.87190869e-02,
-5.37103303e-02, -5.60228005e-02, 2.48959903e-02,
 2.99934857e-02, -7.29027390e-02, -8.02413747e-03,
-6.24387152e-02, -3.78068723e-02, 1.85559671e-02,
-7.56417960e-02, -4.80825827e-02, -4.84495424e-02,
 1.68324355e-02, 4.16389182e-02, -7.51097500e-02,
 2.06383560e-02, 3.08667310e-02, -4.20978293e-02,
 8.39152262e-02, 5.32128289e-02, 8.88345111e-03,
-6.36695698e-02, -1.70899089e-02, 7.27434829e-02,
-5.17251156e-02, 3.84741835e-02, -4.20071259e-02,
-6.81575760e-02, 6.04661182e-02, -9.22805443e-03,
 3.82672697e-02, 5.65126128e-02, -3.89036983e-02,
 7.31798261e-02, -4.94273789e-02, -3.59513015e-02,
-3.02595925e-02, 1.46304090e-02, 2.22913679e-02,
 1.72998998e-02, -7.65312091e-02, -6.85765967e-02,
-6.61176369e-02, 4.36069183e-02, -3.34580392e-02,
-3.46933529e-02, -1.43297995e-03, -7.41180331e-02,
-3.21508162e-02, 8.60543363e-03, -2.08682045e-02,
 7.23989308e-02, 2.06532590e-02, 1.23331156e-02,
 4.27783653e-02, 5.93690090e-02, -2.38721278e-02,
 1.42303074e-03, -5.45618450e-03, 3.83622013e-02,
-2.74501890e-02, 3.98615822e-02, 8.72765761e-03,
 4.82881591e-02, 3.05654351e-02, -7.13570183e-03,
-4.94964123e-02, 4.72280532e-02, 5.84537946e-02,
 3.14183384e-02, 1.03486767e-02, -1.04802232e-02,
-1.63735647e-03, -4.59923781e-02, 1.26902713e-02,
-2.87298467e-02, 5.36087528e-02, 1.57308280e-02,
-9.88585223e-03, 2.72337906e-02, -4.74296138e-02,
 4.19509374e-02, 5.39605543e-02, 5.43201044e-02,
 4.24169786e-02, 6.52010441e-02, -1.35088507e-02,
-6.60623163e-02, 3.28062568e-03, -9.99924820e-03,
 5.70786744e-02, 2.71714535e-02, 9.55113117e-03,
 3.53100225e-02, -8.20715427e-02, 5.38573340e-02,
 3.08814906e-02, -3.77913862e-02, 2.40834262e-02,
 4.63527068e-03, -6.70860782e-02, -8.98967404e-03,
-1.17628893e-03, -1.82394832e-02, -2.22875066e-02,
-6.29196241e-02, 3.24424542e-02, -2.35490054e-02,
 8.47429445e-04, -1.40457144e-02, -3.02314269e-03,

-1.15004499e-02, -5.34413904e-02, 6.05371222e-02,
1.08861467e-02, 7.14789852e-02, 3.18651199e-02,
4.61934358e-02, -2.34730188e-02, -3.03319506e-02,
4.73444946e-02, 1.36363102e-04, -6.25346154e-02,
6.98543862e-02, -6.11930015e-03, 7.91602209e-03,
-2.85243411e-02, 6.94121271e-02, 4.69791563e-03,
5.46652451e-02, -1.20742002e-03, -2.63714977e-02,
5.43937758e-02, 3.86227258e-02, 5.60691301e-03,
3.66772637e-02, -4.42057736e-02, 9.33215662e-04,
1.11030852e-02, -4.93003987e-02, -7.38431364e-02,
1.98967960e-02, -6.97528757e-03, 5.60492836e-02,
-4.58661057e-02, 7.83373937e-02, 3.77670042e-02,
-6.86298404e-03, -3.40715386e-02, -5.88619038e-02,
-3.95905674e-02, 7.69095197e-02, -3.28204110e-02,
5.07092178e-02, 1.35608548e-02, -3.61449234e-02,
2.57839379e-03, 6.47079125e-02, 2.00572591e-02,
-2.51432173e-02, 5.81154041e-02, -5.60516939e-02,
-3.66739407e-02, -6.25331253e-02, -4.21309732e-02,
-1.06909201e-02, -3.75567004e-02, 3.88978864e-03,
-2.97709946e-02, 4.94130924e-02, -6.87331334e-02,
-7.75815267e-03, 3.08784153e-02, 2.95641776e-02,
-7.65760839e-02, 2.03632135e-02, 2.62680948e-02,
-6.36869445e-02, 4.13510837e-02, -3.19908448e-02,
-5.17700836e-02, 6.10096864e-02, 3.14794406e-02,
2.72358172e-02, -8.24764296e-02, 2.95038950e-02,
-5.46698980e-02, 1.79451536e-02, 2.54003778e-02,
-2.18388997e-02, 5.46087623e-02, -5.63516952e-02,
-4.26888391e-02, 4.77498658e-02, 6.15021437e-02,
-7.12481439e-02, -3.63706239e-02, -1.75294615e-02,
5.13981767e-02, -4.99228127e-02, -5.23707457e-02,
-3.18929926e-02, -1.77927688e-02, 2.71088481e-02,
-4.25359420e-02, -3.07249576e-02, 5.66198602e-02,
3.88632119e-02, 3.44969844e-03, -2.18680017e-02,
-6.68943748e-02, 7.62955993e-02, 3.57577056e-02,
-3.14512067e-02, 1.67503543e-02, 1.66078322e-02,
2.12776512e-02, -7.20001832e-02, 6.53690696e-02,
3.19075845e-02, 5.24130613e-02, 5.57781346e-02,
4.72217351e-02, 4.80271019e-02, 3.22762392e-02,
-3.84137058e-03, 1.61821786e-02, -5.46535850e-02,
-1.43720191e-02, 3.42939682e-02, -2.86127385e-02,
-2.36832034e-02, -5.60127944e-02, -6.10959455e-02,
-2.97244824e-02, 6.85454756e-02, -2.30387114e-02,
-7.37300515e-02, -5.91131523e-02, 5.31503931e-02,

1.67027712e-02, -2.99844239e-02, 6.35026768e-02,
-2.79543325e-02, -8.90214592e-02, -7.06197834e-03,
5.04954010e-02, -4.77542616e-02, -1.52074946e-02,
-3.98260579e-02, -2.11178511e-02, -4.98152748e-02,
-7.76334330e-02, 1.65050775e-02, 5.22653647e-02,
7.13527724e-02, -7.11789355e-03, 1.62138119e-02,
2.10232399e-02, -1.64836701e-02, 5.42705990e-02,
7.15925097e-02, 4.50857868e-03, 6.93626469e-03,
-2.78988089e-02, 5.58038242e-02, 3.70827429e-02,
-4.70765196e-02, -5.07270684e-03, 4.93941270e-02,
-2.92939581e-02, 5.31235673e-02, 4.13762257e-02,
7.28104962e-03, -5.16016185e-02, -3.37685016e-03,
-4.19675112e-02, 5.05430996e-02, 6.60187826e-02,
6.37482256e-02, -3.60685103e-02, 6.09117299e-02,
3.87993716e-02, -5.63875437e-02, 4.31440286e-02,
8.86925161e-02, -2.67599989e-03, 5.33812121e-02,
7.24008754e-02, -4.22353484e-02, 1.49651535e-03,
-3.90750729e-02, 5.18729948e-02, -2.20429357e-02,
4.95222285e-02, 5.27303666e-02, -8.37195292e-02,
-7.84929469e-03, -7.57846311e-02, -3.94027270e-02,
-3.67102697e-02, -7.45121110e-03, -5.21211326e-02,
2.79556704e-03, -4.22933288e-02, 3.08283046e-02,
2.38162633e-02, -2.12860163e-02, 3.65846865e-02,
-2.57159118e-02, 2.09319778e-02, -7.85991997e-02,
2.66195163e-02, 1.79166533e-02, -6.51985705e-02,
-6.82042763e-02, 8.99120271e-02, 7.75623322e-02,
-1.76809896e-02, 3.11972909e-02, -7.45557472e-02,
-3.95072922e-02, -5.30850329e-02, -8.51945356e-02,
-6.33281562e-03, 4.11316790e-02, -3.66501845e-02,
1.11192195e-02, -4.05873582e-02, -5.93104064e-02,
-9.75257996e-03, 2.40277797e-02, 6.74796151e-03,
-2.07747109e-02, -6.52206168e-02, -3.28555442e-02,
-2.54805647e-02, -2.45467257e-02, 6.43500760e-02,
2.41759177e-02, -8.69604945e-02, 8.17150027e-02,
-1.94671918e-02, -4.87850159e-02, -1.47961825e-03,
6.30014688e-02, -5.71704879e-02, -4.98677939e-02,
-3.46806720e-02, -2.22330410e-02, -2.68344730e-02,
-4.84029204e-02, -4.66446579e-02, 5.81321195e-02,
1.91990864e-02, 5.23152649e-02, -7.64844567e-02,
3.96070480e-02, 1.80238001e-02, 4.34656553e-02,
-2.62764543e-02, 2.47048568e-02, -4.73491624e-02,
7.94299785e-03, 1.14619834e-02, -3.19231860e-02,
4.07552868e-02, 4.36710455e-02, -9.54134762e-03,

```
 2.20388938e-02, -4.49400097e-02,  8.84186029e-02,
-8.03865865e-02, -5.09085171e-02, -4.69357036e-02,
-8.50972347e-03, -2.16742028e-02, -3.33936177e-02,
-2.71696839e-02, -6.15501441e-02, -3.64925489e-02,
 7.68659711e-02,  4.20241915e-02, -4.65958938e-02,
 1.66038070e-02,  3.67296711e-02,  3.06589399e-02,
 7.49184797e-03, -1.20831269e-03, -2.44503748e-02,
 5.95079400e-02, -4.05839644e-02,  4.16586101e-02,
-4.48176302e-02, -8.39969551e-04, -3.57062905e-03,
 3.81355844e-02, -8.09085369e-02,  4.46945578e-02,
 5.22404052e-02,  1.87518653e-02, -4.48181443e-02,
 1.11654345e-02, -4.93851537e-03,  3.98222394e-02,
 5.72374538e-02,  1.63499955e-02,  6.67956844e-02,
-3.82575253e-03, -4.85672951e-02, -5.82518131e-02,
-7.74941072e-02, -2.63793468e-02,  2.29722373e-02,
-1.45865362e-02, -7.47692659e-02,  3.81551385e-02,
-2.47557294e-02, -8.07667077e-02,  2.98430957e-02,
 4.88052331e-02,  5.78578748e-03,  2.37273369e-02,
-7.34921321e-02, -3.03469673e-02,  5.27568012e-02,
-3.96769523e-04, -3.09300087e-02,  6.25722781e-02,
-8.24268535e-02, -8.42091814e-03, -1.34676704e-02,
-5.50207645e-02, -1.09023219e-02,  3.19536589e-02,
 5.75902127e-02,  2.38687280e-05,  4.32947800e-02,
-3.60564925e-02, -4.71004471e-02,  3.87660600e-02,
-1.17829349e-02,  1.51827966e-03, -1.98617596e-02,
-2.31902320e-02, -6.27214387e-02, -5.74542657e-02,
-6.33974560e-03, -6.12119325e-02, -2.77569443e-02,
-6.69202358e-02, -2.13255063e-02, -1.20965932e-02,
-4.85380776e-02,  7.27683306e-02,  7.12661594e-02,
-4.20774780e-02,  1.02074444e-02,  4.94752359e-03,
 6.77476600e-02,  5.58534376e-02,  4.52718921e-02,
 3.70476730e-02,  4.41624112e-02,  4.22378723e-03,
-1.98311098e-02, -5.56787774e-02, -2.48677898e-02,
 2.38218959e-02,  4.12833057e-02,  4.85572442e-02,
 9.40201059e-03,  1.53387440e-02,  3.12474854e-02,
-3.49078141e-02, -2.01254562e-02, -5.47213890e-02,
 6.80133551e-02, -2.14839893e-04]], dtype=float32)>
```

A sequential model was developed that has one input layer and one output layer and other hidden layers. The model was made up of a single input layer, 4 hidden layers and an output layer. The model was arrived at after a number of variation of various parameters to attain an optimum performance. Parameters that were varied during model development include; hidden layers, dropouts, activation functions, regularizer, epochs, batch and batch size to achieve optimal performance.

The input layer was made from a TensorFlow hub model that were trained for greater-than-word length text, like sentences, phrases or short paragraphs. It consumed human level language text and using the trained set, converted the text into a 512 dimensional vector. A deep learning model was adopted where the input layer fed its output on hidden layer for a deep learning model with number of hidden layers varied to determine the model optimal performance. The layers enable the network to learn complex tasks and achieve excellent performance. The larger the number of hidden layers, the longer time it took for the model to learn. The number of hidden layers was varied before reaching the optimal value of 4 layers. Less or more than four layers led to a reduced accuracy of below 70 %. For instance, a case of 5 hidden layers gave an accuracy of 69%. The last hidden layer output was passed to the output layer that mapped the output into a vector of the estimates for the cleaned question in each individual class. Below is a sample clean question and vector for the output.

Clean question: list reference parameters set hour function

Output vector: [0.57577574 0.506181   0.86234754 0.03550225 0.99626434 0.50395316]

Each figure in the output vector is a weight that corresponds to the below labels respectively.

Class label: [' Analysis ', ' Application ',' Comprehension ', ' Evaluation ', ' Knowledge ', 'Synthesis']

Based on the output, the question is therefore classified as ' Knowledge ' having the highest weight.

An activation function decides whether the neurons are activated or not. In our model development we adopted the use of activation function as we were solving a non-linear problem as opposed to linear transformation. We used relu and sigmoid activation functions in hidden layers and output layer respectively to add non-linearity to the model. Dropouts, techniques of regularization, were adopted to ignore hidden or visible units in artificial neural network model to prevent over-fitting. The final model does not have dropouts after realizing that the model was performing better without them.

The model made use of L1 regularizer to reduce the errors by fitting the function appropriately on the training set and avoid overfitting. The regularizer was varied with L2 and L1_L2 with optimal performance realized with L1. The model development process as well varied the batch and batch size which defines the number of samples to work through before updating internal model parameters. The figures were varied before arriving at an optimal performance. The last item was the epoch that determine the number of cycles that the data was passed during the training. This was varied from as low as 10 to as great as 500 to before estimating the optimal performance range. Figure 4.9 is a neural network model design that was implemented.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer (KerasLayer)    (None, 512)               256797824

 dense (Dense)               (None, 256)               131328

 dense_1 (Dense)             (None, 144)               37008

 dense_2 (Dense)             (None, 72)                10440

 dense_3 (Dense)             (None, 36)                2628

 dense_4 (Dense)             (None, 6)                 222

=================================================================
Total params: 256,979,450
Trainable params: 256,979,450
Non-trainable params: 0
_________________________________________________________________
```

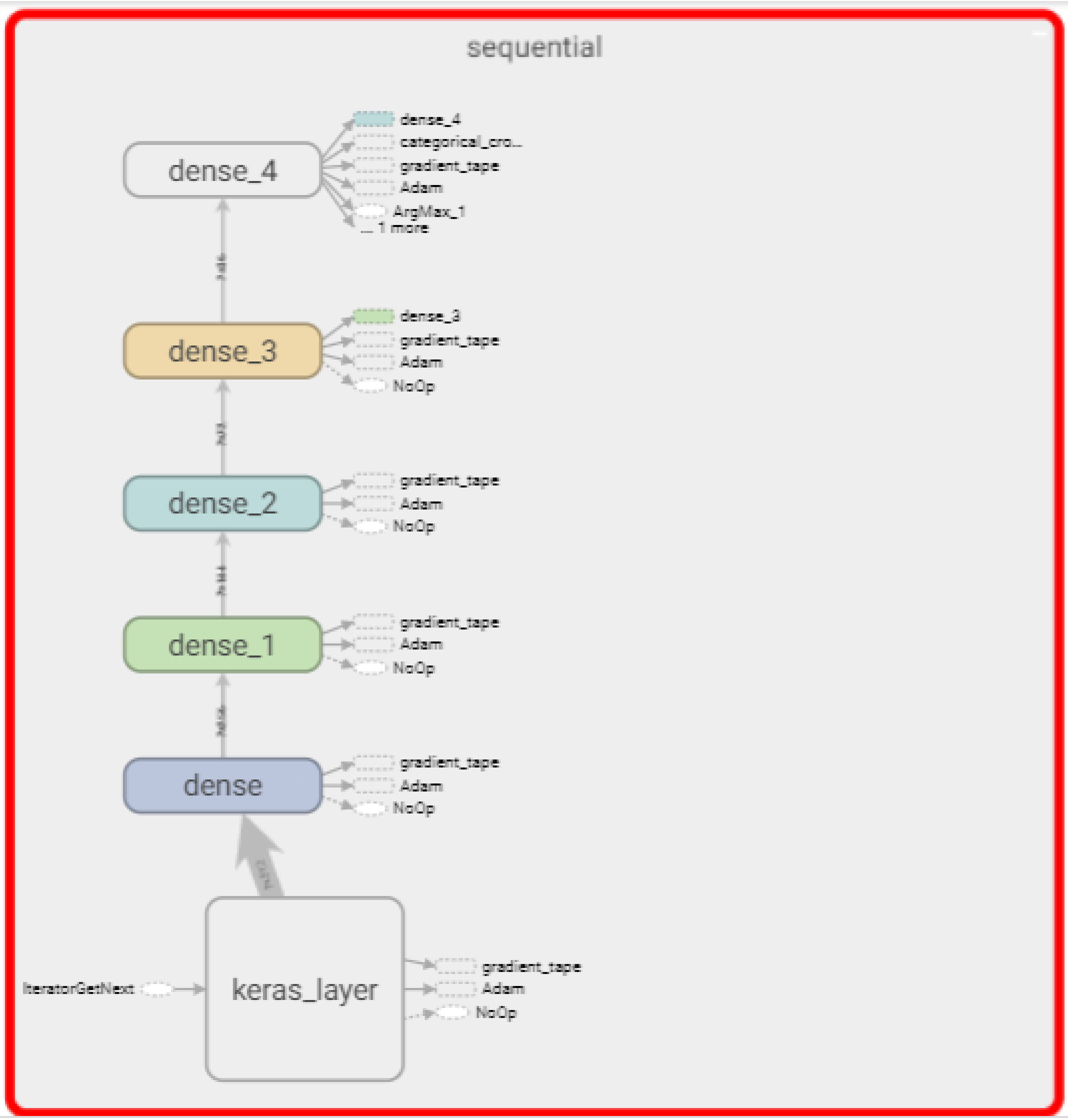

Figure 4. 9: Sequential Model for Neural Network Model

The model was subjected for training and validation by passing it to the respective datasets. The process was done by varying the number of epochs, number of questions shuffled, batch, and batch size until an optimum level of training was realized for this model with the optimal values as 150, 100, 128, and 32 respectively. Figure 4.10 is an output for training where the model was able to learn to a value categorical accuracy of approximate 100% and a value categorical accuracy estimate of 73%. Categorical accuracy was used to measure the performance of the model on the test data while the value categorical accuracy was used to measure the performance on the validation data. Our model achieved a categorical accuracy of 100% and value categorical accuracy of 73% as illustrated in figure 4.10.

```
4/4 [==============================] - 10s 2s/step - loss: 5.8042 - categorical_accuracy: 1.0000 - val_loss: 6.2896 - val_categorical_accuracy: 0.7333
Epoch 39/50
4/4 [==============================] - 10s 2s/step - loss: 5.3952 - categorical_accuracy: 1.0000 - val_loss: 5.8934 - val_categorical_accuracy: 0.7400
Epoch 40/50
4/4 [==============================] - 10s 2s/step - loss: 5.0055 - categorical_accuracy: 1.0000 - val_loss: 5.5052 - val_categorical_accuracy: 0.7333
Epoch 41/50
4/4 [==============================] - 11s 3s/step - loss: 4.6307 - categorical_accuracy: 1.0000 - val_loss: 5.1502 - val_categorical_accuracy: 0.7467
Epoch 42/50
4/4 [==============================] - 10s 2s/step - loss: 4.2767 - categorical_accuracy: 1.0000 - val_loss: 4.8081 - val_categorical_accuracy: 0.7400
Epoch 43/50
4/4 [==============================] - 10s 2s/step - loss: 3.9465 - categorical_accuracy: 1.0000 - val_loss: 4.4862 - val_categorical_accuracy: 0.7267
Epoch 44/50
4/4 [==============================] - 10s 2s/step - loss: 3.6379 - categorical_accuracy: 1.0000 - val_loss: 4.1891 - val_categorical_accuracy: 0.7467
Epoch 45/50
4/4 [==============================] - 10s 2s/step - loss: 3.3474 - categorical_accuracy: 1.0000 - val_loss: 3.9088 - val_categorical_accuracy: 0.7333
Epoch 46/50
4/4 [==============================] - 10s 2s/step - loss: 3.0714 - categorical_accuracy: 1.0000 - val_loss: 3.6436 - val_categorical_accuracy: 0.7333
Epoch 47/50
4/4 [==============================] - 10s 2s/step - loss: 2.8169 - categorical_accuracy: 1.0000 - val_loss: 3.3961 - val_categorical_accuracy: 0.7267
Epoch 48/50
4/4 [==============================] - 10s 2s/step - loss: 2.5846 - categorical_accuracy: 1.0000 - val_loss: 3.1759 - val_categorical_accuracy: 0.7267
Epoch 49/50
4/4 [==============================] - 11s 3s/step - loss: 2.3696 - categorical_accuracy: 1.0000 - val_loss: 2.9755 - val_categorical_accuracy: 0.7267
Epoch 50/50
4/4 [==============================] - 10s 3s/step - loss: 2.1730 - categorical_accuracy: 1.0000 - val_loss: 2.7921 - val_categorical_accuracy: 0.7333
```

Figure 4. 10: Model Training Output

The loss which is the distance between ground truth and prediction that our network model is trying to reduce for is as illustrated in figure 4.11

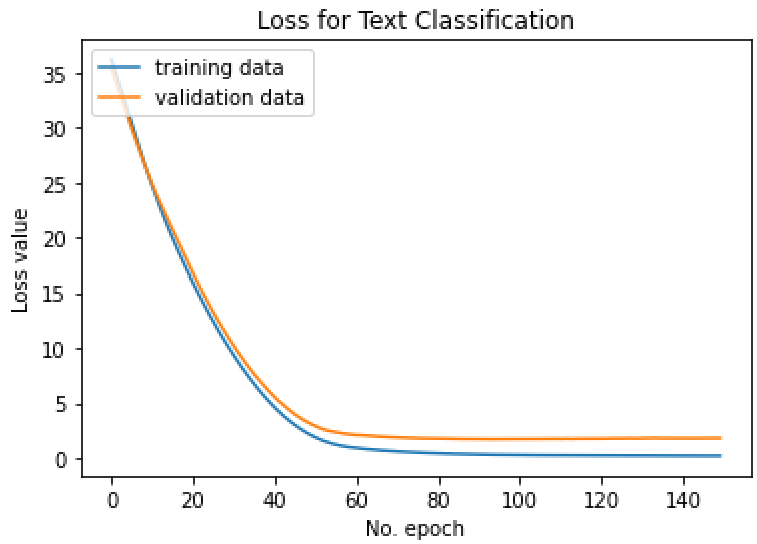

Figure 4. 11: Loss for Question Classification Model

The model accuracy which gives the percentage of instances that are correctly classified was plotted as illustrated in the figure 4.12

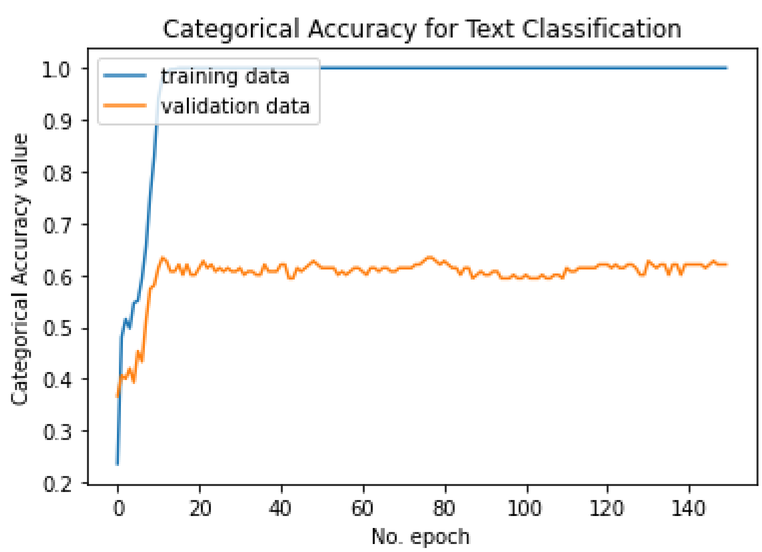

Figure 4. 12: Categorical Accuracy for Question Classification

The two figures indicate that the model is performing properly on the validation set and therefore has a high categorical accuracy. The model was then evaluated using the test dataset with a sample vector output for the test dataset as follows;

```
[[3.97804379e-03 6.46262228e-01 9.45989132e-01 8.11931491e-02
  9.93797481e-01 8.12512636e-01]
 [1.42379884e-06 9.53891039e-01 9.98940229e-01 1.73722208e-02
  7.09709764e-01 6.07306957e-01]
 [2.29482248e-05 9.86297131e-01 5.49318552e-01 1.76842362e-01
  2.53358483e-01 7.18128443e-01]
 [8.95474076e-01 6.22648716e-01 1.01500750e-03 9.92354989e-01
  7.88802147e-01 9.54128444e-01]
 [4.14745390e-01 2.22076267e-01 2.12809235e-01 8.20457637e-02
  3.45541298e-01 9.89785433e-01]
 [8.23191941e-01 8.82327616e-01 4.16159630e-04 9.99401212e-01
  5.82477987e-01 7.37570941e-01]
 [3.70889902e-04 9.67283368e-01 3.26608837e-01 5.62182665e-01
  5.53810656e-01 7.22163081e-01]
 [1.50740152e-05 9.71052647e-01 9.14412141e-01 4.07918692e-02
  3.62740666e-01 7.51771927e-01]
 [1.71299785e-01 6.22771621e-01 3.53547037e-02 6.17838919e-01
  4.84398872e-01 9.72841859e-01]
 [9.99891877e-01 3.92098427e-02 5.03331423e-04 9.83303308e-01
  9.50005889e-01 9.93721306e-01]
 [3.94409597e-01 6.61617339e-01 7.82024860e-03 8.67960751e-01
  3.44933689e-01 9.69589114e-01]
 [9.25819397e-01 8.17503631e-01 3.18288803e-04 9.99355435e-01
  6.94561720e-01 8.09299111e-01]
```

Each array in the vector correspond with output for each question which indicates the weights for every class. The respective predictions run from 0 to 5 with every integer value indicating the position of each question's class with the weights as provided by our output. For instance, the first array in our output has a prediction of 4 which translate to the 5th class in our class labels which is 'Knowledge ' as follows;

[4 2 1 3 5 3 1 1 5 0 5 3 4 1 1 3 5 3 4 1 1 4 5 5 4 2 1 4 5 0 4 2 3 0 5 3 4
1 1 0 5 3 4 2 5 0 1 3 4 2 5 0 5 3 2 2 5 5 5 3 4 5 2 0 5 3 4 5 1 1 5 3 4 2
2 5 3 3 4 4 5 0 5 3 4 2 3 3 5 3 4 1 4 5 3 4 2 1 5 3 4 5 5 2 3 4 2 0 5 3 4
4 0 5 5 4 2 3 5 3 2 4 3 5 3 4 2 4 5 3 4 5 3 4 5 4 1 5 1 5 5]

Class label: [' Analysis ', ' Application ',' Comprehension ', ' Evaluation ', ' ***Knowledge*** ', 'Synthesis']

The model testing for the test questions is as shown in figure 4.13. The class column is the true class label while the pred_sentiment column is the predicted class. This shows that 10 out of the 13 questions classes were correctly predicted.

	Sno	Class	Question	Num_words_text	pred_sentiment
0	1	Knowledge	list reference parameters sethour function	8	Knowledge
1	1	Comprehension	explain what happening what meant	15	Comprehension
2	1	Application	design sketch marketing strategy your product ...	16	Application
3	1	Analysis	analyze selected information	4	Application
4	1	Synthesis	create several different strategies solve math...	9	Synthesis
5	1	Evaluation	defend your position about	8	Evaluation
6	2	Knowledge	explain briefly meaning following terms demogr...	16	Comprehension
7	2	Comprehension	class facility abstract class explain what thi...	10	Comprehension
8	2	Application	sketch experiment plants grow different kinds ...	13	Analysis
9	2	Analysis	discriminate difference parts	10	Analysis
10	2	Synthesis	propose alternative plan	11	Synthesis
11	2	Evaluation	assess value importance	12	Evaluation
12	3	Knowledge	label parts diagram	6	Knowledge

Figure 4. 13: Model evaluation on the test data

Precision, recall and accuracy are three metrics that are used to measure the performance of a machine learning algorithm. The model effectiveness was measured by determining model precision, recall, f1-score, support, accuracy, macro average and weighted average. Model accuracy recorded 71% performance with metrics as shown in the figure 4.14.

	precision	recall	f1-score	support
Knowledge	0.79	0.88	0.84	26
Comprehension	0.67	0.70	0.68	23
Application	0.41	0.47	0.44	15
Analysis	0.76	0.57	0.65	23
Evaluation	0.79	0.79	0.79	24
Synthesis	0.73	0.73	0.73	30
accuracy			0.71	141
macro avg	0.69	0.69	0.69	141
weighted avg	0.71	0.71	0.71	141

Figure 4. 14: Model Precision and Recall

From the figure 4.14, the model effectiveness shows that it performs a bit poor on the application questions. This implies that the examiners may need to be keen on applications questions when using the model

4.3.2 Questions Classification Application Programming Interface (API)

An API is a product that allows services and products to communicate with each other and leverage each other's data and functionality through a documented interface. An API works by client application making a request in a certain predefined order to the web server by the use of uniform resource identifier (URI). On receiving the request, the API makes query from the webserver and obtain response sent by the server to provide as an output for the client app. This request contains a request verb, headers, and sometimes, a request body.

Following design and implementation of the model through training using the neural network algorithm an API was developed to allow classification of questions based on the model. The API was designed to take a question as an input and using the saved model determine question's classification. The model provides the question's class as the output. The API was developed using *python language and Django-rest framework*. Figure 4.15, are diagrams showing how the request is sent, the output received and a figure 4.16 showing the processing logs on the backend.

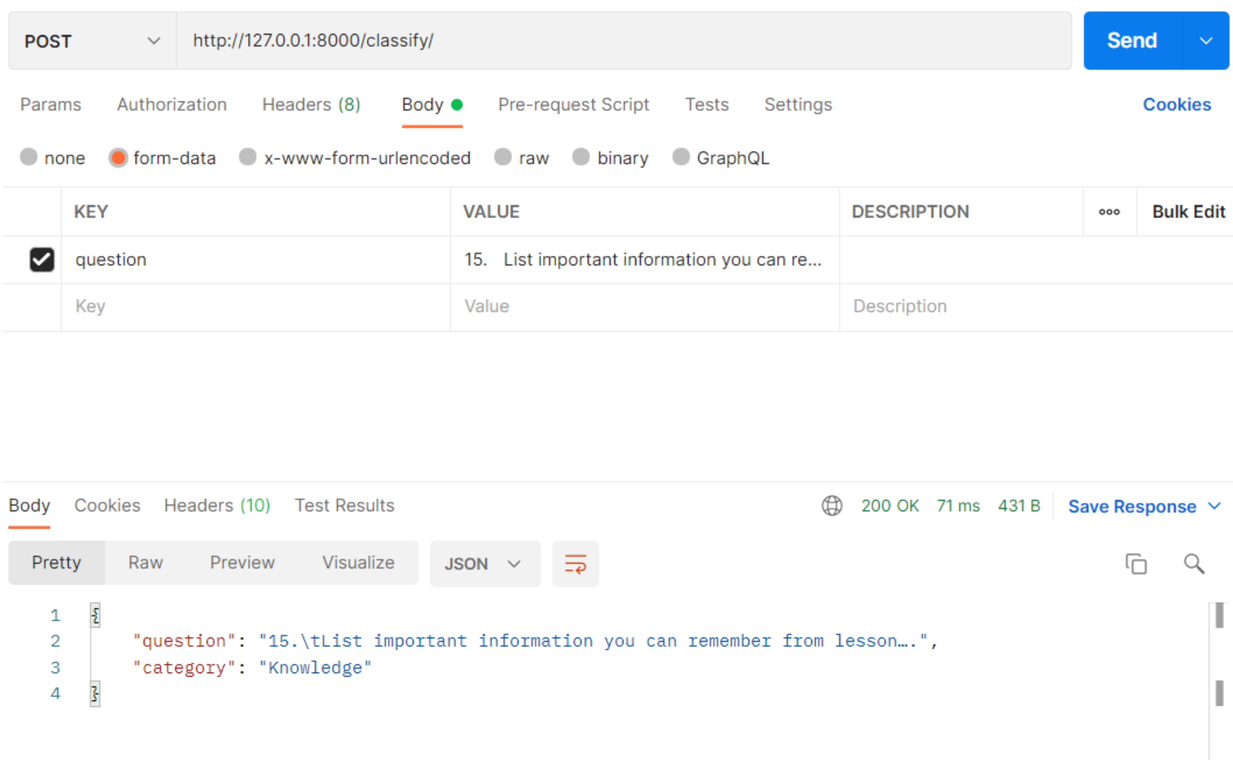

Figure 4. 15: Sending and Receiving Requests through Postman

In the figure 4.15, a get request was sent containing a question No. 15 "List important information you can remember from lesson ….." in a request body. The API process is as shown in figure 4.16.

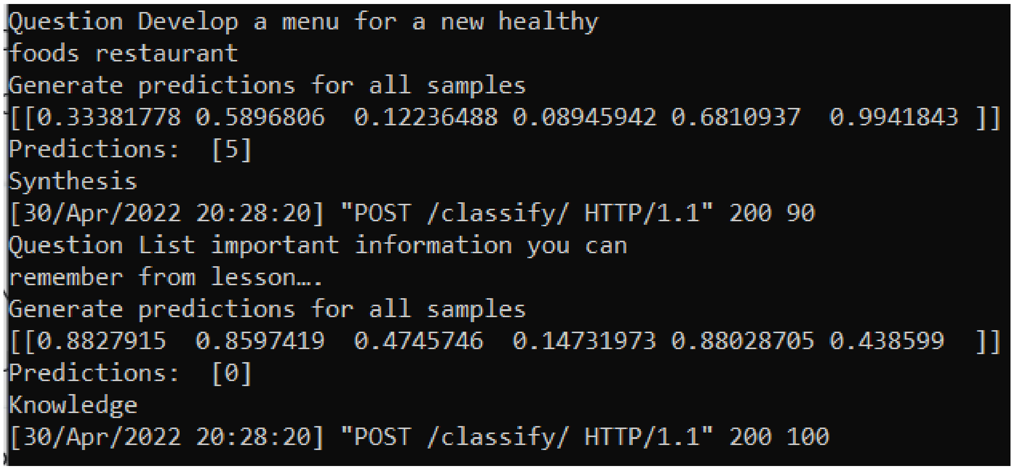

Figure 4. 16: Model Processing and Output through CMD

The API processes the request by submitting the question to the webserver where the question classifier model is used to estimate the question class and provide the class label. This label was then sent via API to the client caller as shown in the figure 4.17.

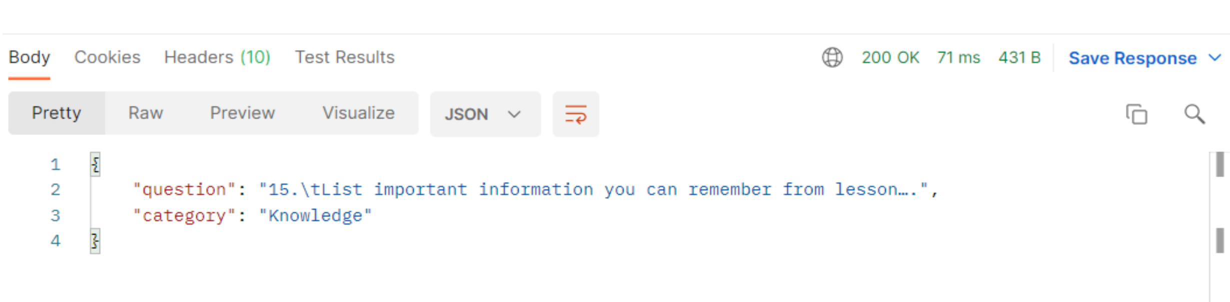

Figure 4. 17: API response to the client request

4.3.3 Integration with Learning Management System (LMS)

The Moodle LMS was adopted for integration with the classifier as the LMS is an open source that allow extending its functionality through plugins and as well offer online web-based automated testing by the use of quizzes. A plugin was developed using the guidelines for Moodle development and provided for use to automate questions classification in the LMS units. The language used was the *PHP* as it is the language used in core Moodle LMS development and its plugins. The plugin was developed using three stages namely: analysis, design, and development.

The first stage was to analyze LMS structure and how the questions are represented and stored. This yielded the fact that the LMS has a method to define and store a question. Beside this, the LMS use tags as a method to classify questions. Moodle question tags are defined during questions creation or later in the question bank. The second was to internalize and come up with a design where, the tags in the LMS can be generated for the questions classification using their cognitive levels. This design factored the fact that the tags played a greater role especially during setting questions for quizzes where a questions are picked based on their classification.

The last stage was to develop the plugin and integration to the LMS. Moodle plugin development tutorials were heavily used to guide on the baseline. The plugin was developed to detect questions in a particular course that are not classified, alert the privileged user and allow them to click and classify the unclassified questions in the course. The plugin was developed to send the request to the API and use the class returned to classify the questions in Moodle into the six (6) levels of Bloom's Taxonomy. Figure 4.18 that shows some questions that have not been categorized in Moodle which are marked in yellow colored braces.

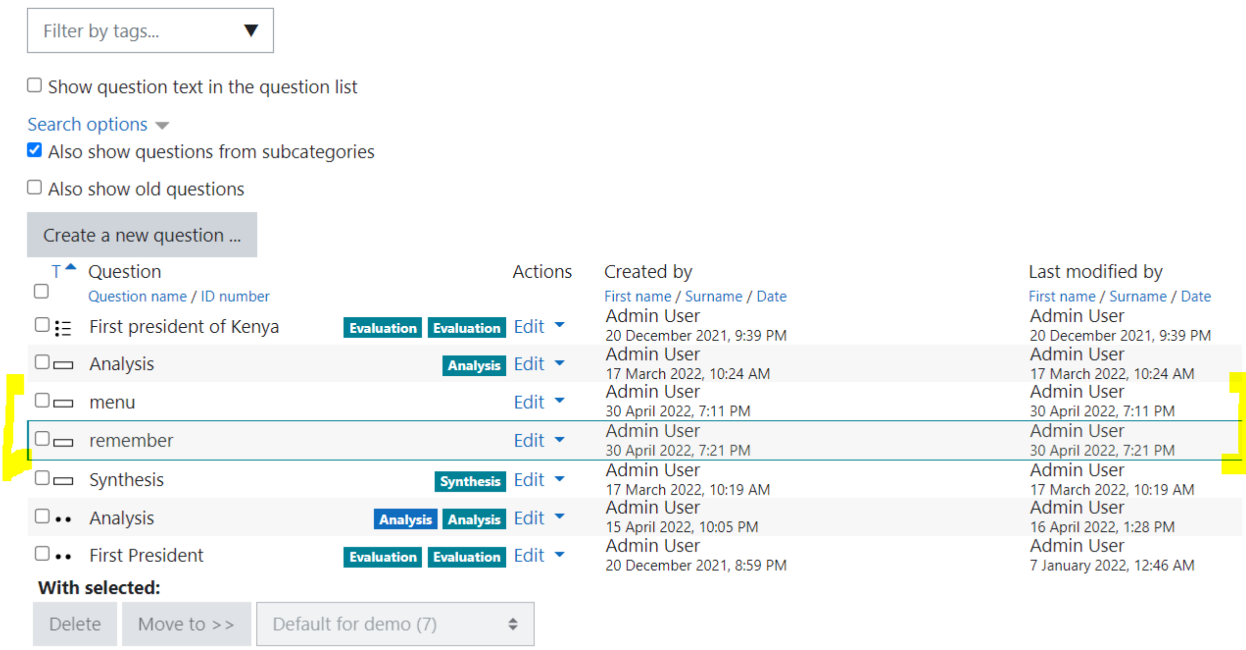

Figure 4. 18: Questions to be Classified in the Question Bank

The plugin detects new questions and alerts the teacher or any other user with privileges on the need to classify newly added questions. The figure 4.19 shows the alert message shown when new questions that are yet to be classified are available.

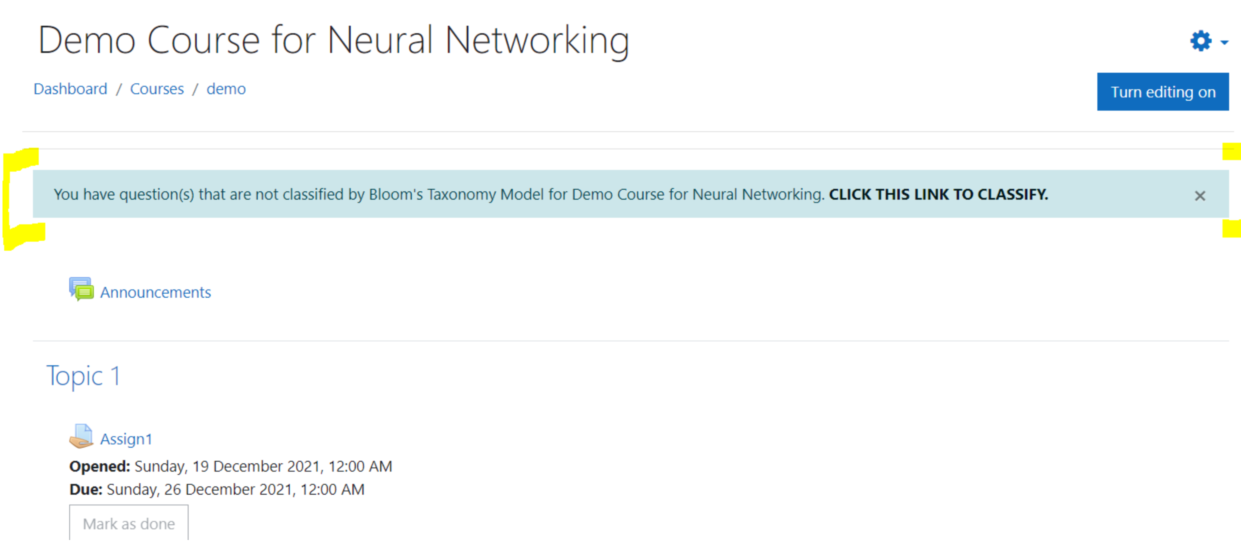

Figure 4. 19: Alert to the Privileged User

When "CLICK THIS LINK TO CLASSIFY" link is clicked, questions in the course that are not classified are classified and the output for the classed shown to the user as shown in the figure 4.20.

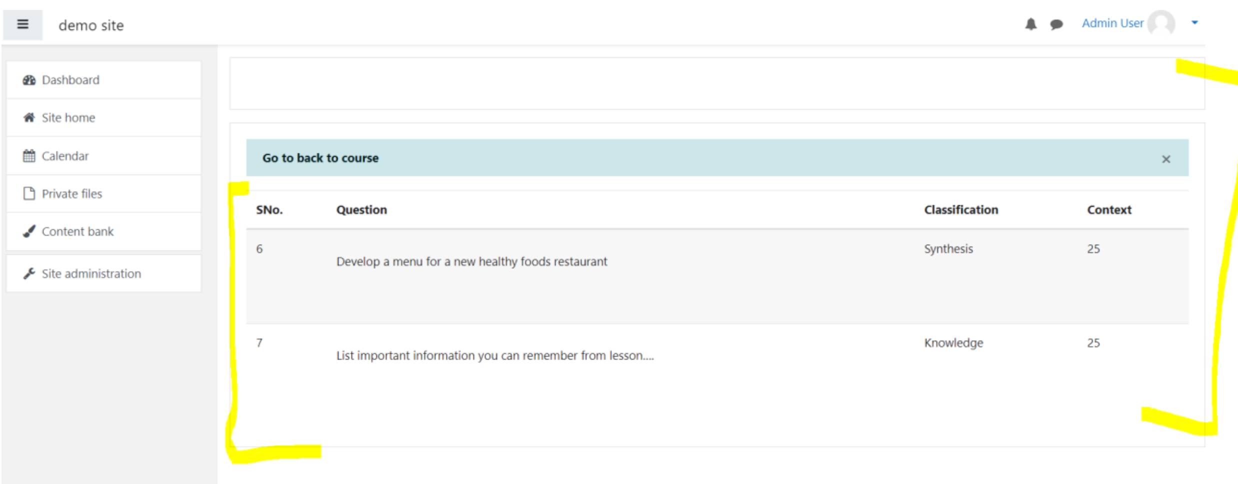

Figure 4. 20: Results for Question Classification

The questions when viewed again gives the below output.

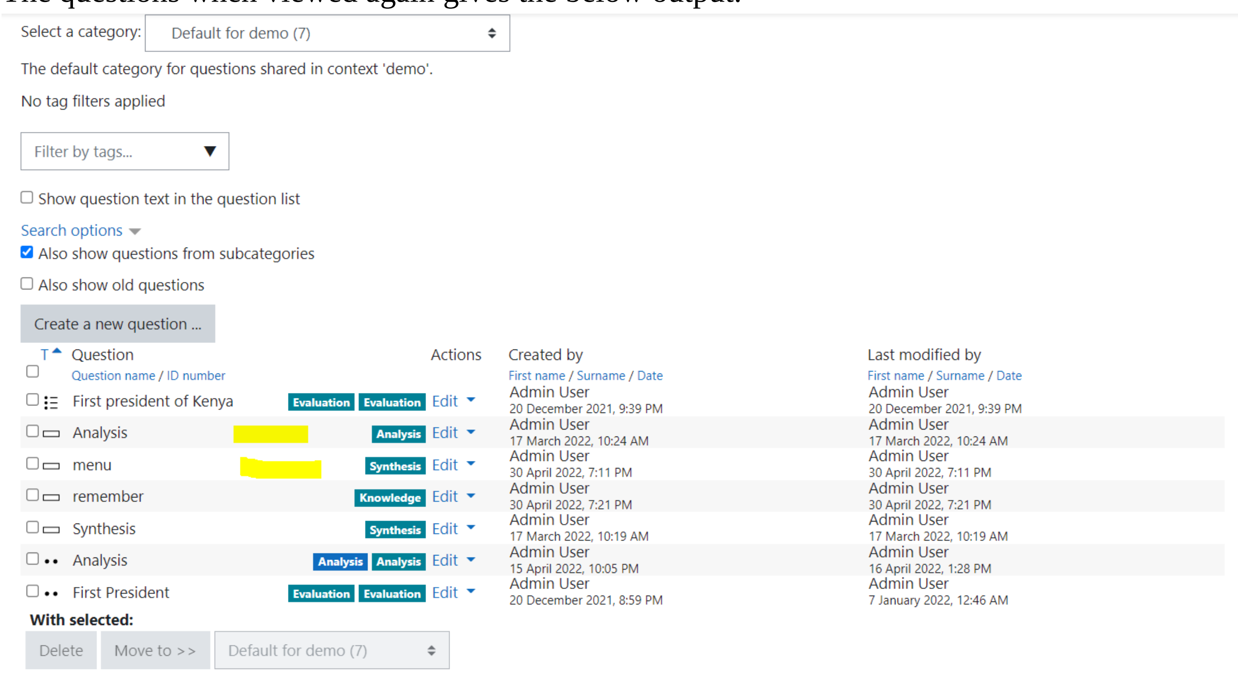

Figure 4. 21: Classified Questions in the Question Bank

4.4 System Testing

This refers to the process of putting a fully integrated software system to the test. The system underwent unit, integration and usability tests.

4.4.1 Unit testing

This involved testing each unit of the system proposed autonomous. The system has 3 major components; model, API, and Moodle plugin. The model was tested using the testing data and gave the output as shown in Figure 4.13. The API was independently tested to establish whether the client request is received and a response provide as shown in the figures 4.15 to 4.17. The Moodle plugin was independently tested to find out whether newly added questions were classified as shown in the figures 4.18 to 4.21.

4.4.2 Integration testing

The model, API and plugin were integrated to see if given a question the plugin was able to call the API and a response given to the caller app. Figure 4.19 shows a link "CLICK THIS LINK TO CLASSIFY" which when clicked makes an API request and the response is as shown in figure 4.21.

4.4.3 Usability Testing

The system was given a list of unlabeled questions, classified them and stored them in a database. The classified questions form part of random questions that are candidate for selection during exam generation. A sample quiz made from the use of the question label was as shown in figure 4.22.

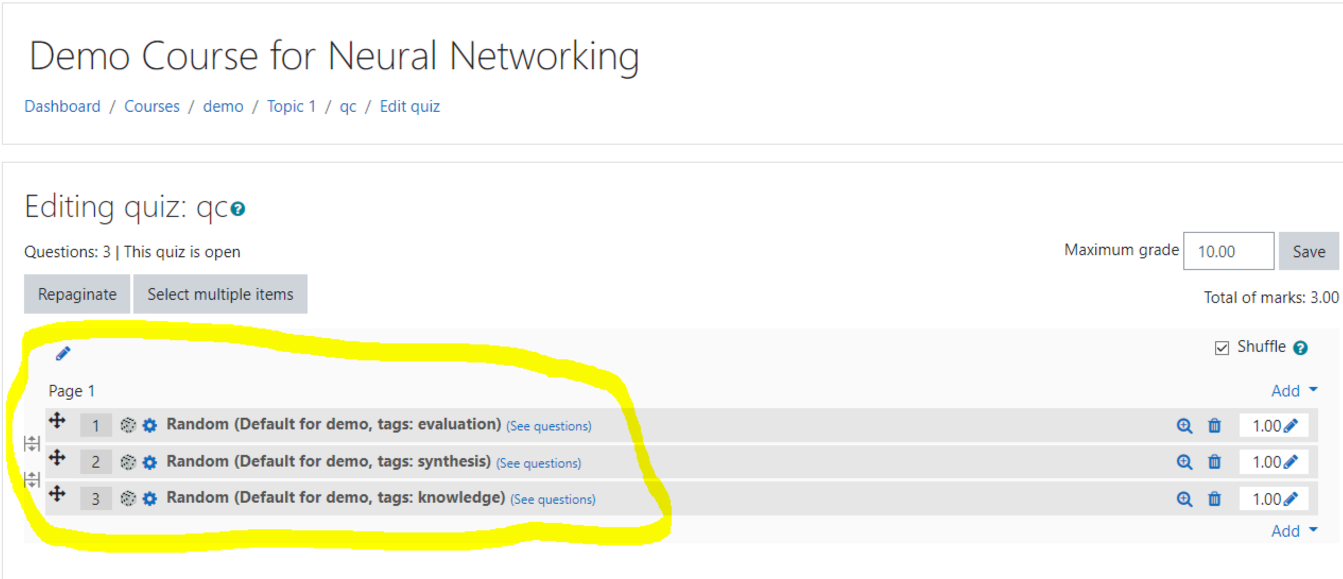

Figure 4. 22: Sample quiz based on the questions classification labels

4.5 Discussion

This research has been able to explore the use of various technologies in the development of questions and examination system. This research has made the use of a deep learning algorithm, ANN by the help of NLP to produce a question cognitive level classifier based on the Bloom's Taxonomy. The model was trained using a set of classified data, validated and tested with a new set of data, which recorded 71% accuracy. Comparing our results with other algorithms that have been studied in other research we provide the following observations (Mohammed & Omar, 2020):

- KNN algorithm for the collected dataset gave a precision of about 65% to 78% with 79% to 87% for the evenly distributed data that was in large amount.
- LR algorithm for the collected dataset gave a precision of about 71% to 86% with 83% to 90% for the evenly distributed data that was in large amount.
- SVM algorithm for the collected dataset gave a precision of about 77% to 86% with 84% to 90% for the evenly distributed data that was in large amount.

A comparison of SVM and K-NN was performed, the former performed well with 92.3 % compared to the latter with 66.6%. This indicated that SVM was a better learner in their experiment(Patil & Shreyas, 2018).

There was a satisfactory performance of SVM that declined with the increase in the amount of words used to represent a question (Yahya et al., 2012).

The research results show that the model developed was within the range that other models provided, however, at a lower level probably due to the limitation of the amount of learning data as deep learning require huge amount of data that was not readily available. The ANN has satisfactory performance that may improve with the quality and amount of questions. The results indicate poor performance for applications questions which could make the accuracy improve if a solution is established on the cause for the particular class. The model performed poorly on this set of data may be due to the quality of the questions in terms the amount of words used to represent them.

The model was tested during development using the testing dataset as well as using the questions developed by lecturers. The testing questions were loaded in the LMS and the API integrated in the system using a plugin. The questions were classified into their cognitive levels that were used to generate an exam for students. It is therefore possible to use ANN for deep learning to develop a question classifying model and use it during exam generation.

CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS

5.0 Introduction

This chapter covers review of research objectives with a discussion on how the research was able to adequately achieve them, a discussion on the findings, recommendations and a highlight to future research directions is discussed.

5.1 Summary of Findings

The literature review focused on automation technologies used in auto-generation of exams. The technologies that were used in the previous research work ranged from experienced to intelligent ones with algorithms such as support vector machine and K-Nearest Neighbour having been used in some projects. The research has helped to establish that institutions were in the process of fully utilizing artificial intelligence in automation of the examination generation process. E-learning introduced flexible learning patterns where automated tests are needed to examine learning. Existing methods for classifying questions cognitive levels were developed with some recommending use of deep learning algorithms. Technologies covered included use of programming languages, filtering techniques, artificial intelligence, randomization among others. The review was informative and formed the basis for machine learning model design as well as the choice of algorithm and technologies deployed in this project.

The research was able to come up with a model for the ANN that consumed the NLP preprocessed output and the resultant model tested using LMS questions. The model was tuned to an optimal accuracy of 71%. Unit integration and other tests were performed to ensure proper model performance.

An API was developed to allow consumption of the model by an LMS. The API accepted GET request for the question and returns a label for the question which is the estimated class or cognitive level. The API was developed and supplied with questions and provide question's category as the output as show in the postman output in figure 4.14. Lastly a plugin was developed to facilitate consumption of the API response by Moodle LMS. This allowed the integration of the model into the LMS. Figure 4.17 provides a view on how questions were classified in the LMS.

Technological advancement has given rise to great techniques that are available for application in questions classification and examination generation. Artificial intelligence enabled automation of questions classification using MLA (Machine Learning Algorithms). A neural network model combined with pre-processor enabled training and testing of the model. The model subjected to data (questions) was able to approximate the question class at an accuracy of about 71%. The model when used on an application using an API was able to classify questions into their respective Bloom's Taxonomy levels. Finally, the model was used during the process of exam setting and testing to generate examination based on the questions cognitive levels. Figure 5.1 below shows how to use the classified questions when setting quizzes.

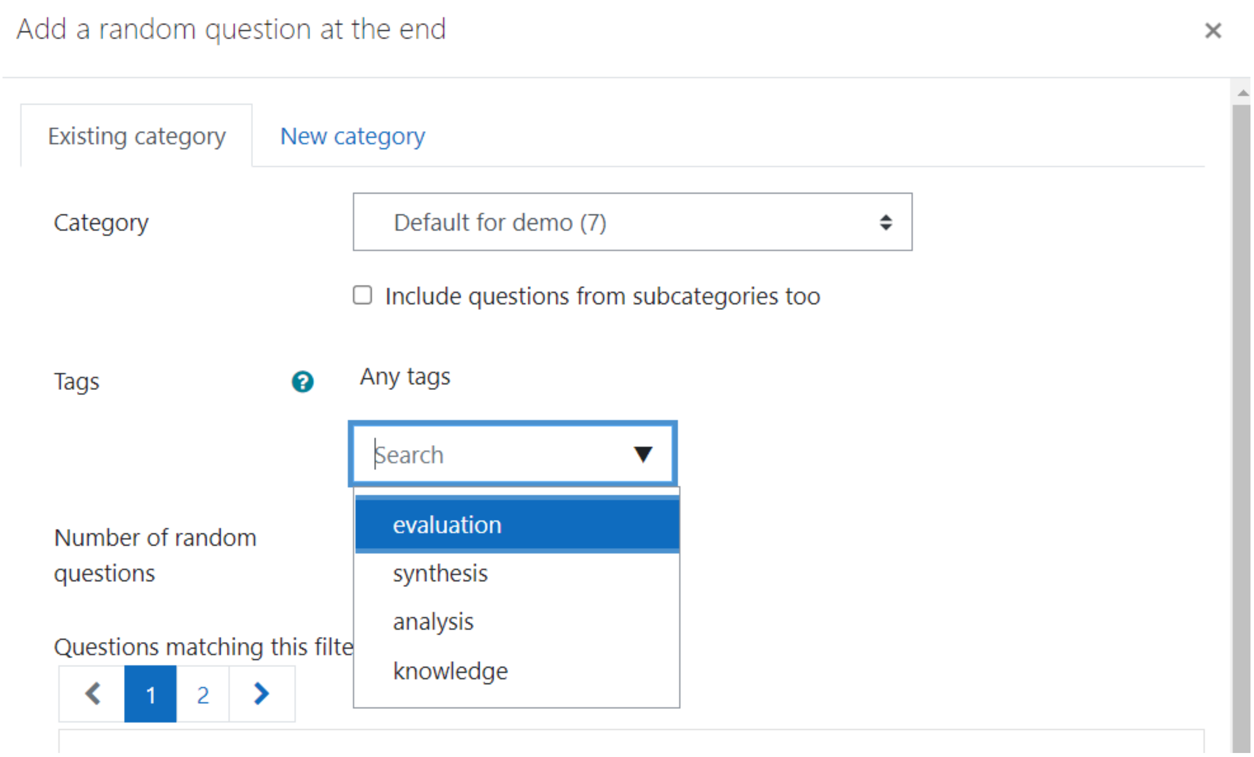

Figure 5. 1: Adding a Question to the Quiz Based on Complexity

5.2 Conclusion

Neural network algorithm was able to classify exam questions for use during testing. The algorithm was used to predict questions cognitive levels based on Bloom's taxonomy. Labelled questions were used for model training and the resultant model testing. Natural language processing was used to convert the question into vectors that a machine understands for model development. The model effectiveness was determined with an accuracy of 71%. This model was able to classify questions according to their labels with application questions having a lower weight compared to others. The API and plugin were able to provide a mechanism for integrating the system with a Moodle LMS for questions classification and testing.

5.3 Recommendations

The model for questions classification has been developed. The model has an accuracy of 71% for question classification. Though it performs a little poor on application questions, the model can be put to use for questions classification through the help of the API to aid the examiners on the questions cognitive levels.

A plugin for Moodle LMS has been developed for use with the API. This can be useful for examiners using Moodle to benefit in the process of determining questions cognitive levels.

5.4 Future Research Direction

Questions classification involve learning the structures in the natural language, this can only happen to questions in a particular language. The most studied language is the English language and its structure is not the same across other languages. To enable the generalization, more libraries that have other languages need to be incorporated.

Deep learning algorithm (ANN) with NLP helping to pre-process text data were used in this project. The NLP was configured to a sequential feed forward setup. Future research can consider use of convolutional reinforcement learning to establish its effectiveness. To establish the best algorithm for classification of questions to their cognitive domain, an empirical evaluation for various algorithms should be done.

REFERENCES

Abduljabbar, D. A., & Omar, N. (2015). Exam questions classification based on Bloom's taxonomy cognitive level using classifiers combination. *Journal of Theoretical and Applied Information Technology*.

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, *4*(11), e00938.

Acemoglu, D., & Restrepo, P. (2019). *8. Artificial Intelligence, Automation, and Work*. University of Chicago Press.

Ali, N. A., Eassa, F., & Hamed, E. (2018). Adaptive E-Learning System Based on Personalized Learning Style. *Journal of Fundamental and Applied Sciences*, *10*(4), 246–251.

Balfe, N., Sharples, S., & Wilson, J. R. (2015). Impact of automation: Measurement of performance, workload and behaviour in a complex control environment. *Applied Ergonomics*, *47*, 52–64.

Buick, J. M. (2011). Physics Assessment and the Development of a Taxonomy. *European J of Physics Education Vol.2 Issue 1 2011 ISSN1309 7202*.

Diab, S., & Sartawi, B. (2017). Classification of Questions and Learning Outcome Statements (LOS) into Bloom's Taxonomy (BT) by Similarity Measurements Towards Extracting of Learning Outcome from Learning Material. *International Journal of Managing Information Technology*. https://doi.org/10.5121/ijmit.2017.9201

Eldesoky, I., Aboutabl, A., & Haggag, M. (2014). Semantic Attributes Model for Automatic Generation of Multiple Choice Questions. *International Journal of Computer Applications*, *103*, 975–8887. https://doi.org/10.5120/18038-8544

Gupta, N. (2013). Artificial neural network. *Network and Complex Systems*, *3*(1), 24–28.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*(6245), 261–266.

Immanuel, A. (2015). Tulasi. B,"Framework for Automatic Examination Paper Generation System,." *International Journal of Computer Science Trends and Technology*, *6*(1).

Joshi, A., Joshi, M., & Doiphode, S. (2016). A survey on question paper generation system. https://www.ijcaonline.org/proceedings/ncrenb2016/number1/25549-4014.

Karamustafaoglu, S., Karamustafaoglu, O., Bacanak, A., & Degirmenci, S. (2011). Classification of biology exam questions as to bloom. *Energy Education Science and Technology Part B: Social and Educational Studies*.

Kumara, B. T. G. S., Brahmana, A., & Paik, I. (2019). Bloom's Taxonomy and Rules Based Question Analysis Approach for Measuring the Quality of Examination Papers. *International Journal of Knowledge Engineering*. https://doi.org/10.18178/ijke.2019.5.1.111

Kurdi, G., Leo, J., Parsia, B., Sattler, U., & Al-Emari, S. (2020). A Systematic Review of Automatic Question Generation for Educational Purposes. *International Journal of Artificial Intelligence in Education*, *30*, 121–204. https://doi.org/10.1007/s40593-019-00186-y

Liddy, E. D. (2001). *Natural language processing*.

Manogharan, M. W., Thivaharan, T., & Rahman, R. A. (2018). Academic Staff Retention in Private Higher Education Institute--Case Study of Private Colleges in Kuala Lumpur. *International Journal of Higher Education*, *7*(3), 52-78.

Meskhi, B., Ponomareva, S., & Ugnich, E. (2019). E-learning in higher inclusive education: needs, opportunities and limitations. *International Journal of Educational Management*.

Mohammed, M., & Omar, N. (2020). Question classification based on Bloom's taxonomy cognitive domain using modified TF-IDF and word2vec. *PloS One*, *15*(3), e0230442.

Mohammedid, M., & Omar, N. (2020). Question classification based on Bloom's taxonomy cognitive domain using modified TF-IDF and word2vec. *PLoS ONE*. https://doi.org/10.1371/journal.pone.0230442

Naeem, M., Rizvi, S. T. H., & Coronato, A. (2020). A gentle introduction to reinforcement learning and its application in different fields. *IEEE Access*, *8*, 209320–209344.

Ndirangu, P. N., Muuro, E. M., & Kihoro, J. M. (2021). A Literature Review on Automatic Generation of Examinations. *Open Journal For*, 77.

Ogula, P. A., Muchoki, F., M., Dimba, M., and Machyo, C.(2006). Practical Guide to Teaching Practice for Students and Lecturers. Catholic University of Eastern Africa: Nairobi.

Omar, N., Haris, S. S., Hassan, R., Arshad, H., Rahmat, M., Zainal, N. F. A., & Zulkifli, R. (2012). Automated Analysis of Exam Questions According to Bloom's Taxonomy. *Procedia - Social and Behavioral Sciences*. https://doi.org/10.1016/j.sbspro.2012.09.278

Patil, S. K., & Shreyas, M. M. (2018). A Comparative Study of Question Bank Classification based on Revised Bloom's Taxonomy using SVM and K-NN. *2017 2nd International Conference On Emerging Computation and Information Technologies, ICECIT 2017*. https://doi.org/10.1109/ICECIT.2017.8453305

Praveena, M., & Jaiganesh, V. (2017). A literature review on supervised machine learning algorithms and boosting process. *International Journal of Computer Applications*, *169*(8), 32–35.

Rahim, T. N. T. A., Domingo, M. S. T., Batcha, M. F. N., & Aziz, Z. A. (2020). Automated Exam Question Set Generator Using Utility Based Agent and Learning Agent. *International Journal of Machine Learning and Computing*. https://doi.org/10.18178/ijmlc.2020.10.1.914

Rao, V. S., Sai, V. C., Sandeep, S. S., & Jayadeep, M. V. (2020). Automated Exam Paper Process Based on Schedule and Authenticity. *Available at SSRN 3643880*.

Shekhar, S. S. (2019). Artificial intelligence in automation. *Artificial Intelligence*, *3085*(06), 14–17.

Soni, V. D. (2019). IOT connected with e-learning. *Vishal Dineshkumar Soni.(2019). IOT Connected with e-Learning. International Journal on Integrated Education*, *2*(5), 273–277.

Tang, X., Chen, Y., Li, X., Liu, J., & Ying, Z. (2019). A reinforcement learning approach to personalized learning recommendation systems. *British Journal of Mathematical and Statistical Psychology*, *72*(1), 108–135.

Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K.-L. A., Elkhatib, Y., Hussain, A., & Al-Fuqaha, A. (2019). Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE Access*, *7*, 65579–65615.

Yahya, A. A., Toukal, Z., & Osman, A. (2012). Bloom's taxonomy-based classification for item bank questions using support vector machines. *Studies in Computational Intelligence*. https://doi.org/10.1007/978-3-642-30732-4-17

Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. *National Science*

Review, *5*(1), 44–53.

APPENDIX

Appendix 1: Code for the Developed Model

```
import matplotlib.pyplot as plt
import os
import re
import shutil
import string
import tensorflow as tf

import tensorflow_hub as hub

from tensorflow.keras import layers
from tensorflow.keras import losses
from tensorflow.keras import regularizers

from collections import Counter


import pandas as pd
import numpy as np

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split

import matplotlib.pyplot as plt


import pydot



#  Define functions Clean the data by removing the emoticons and symbols
def remove_emoji(text):
    emoji_pattern = re.compile("["
                           u"\U0001F600-\U0001F64F"  # emoticons
                           u"\U0001F300-
\U0001F5FF"  # symbols & pictographs
                           u"\U0001F680-
\U0001F6FF"  # transport & map symbols
                           u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                           u"\U00002702-\U000027B0"
                           u"\U000024C2-\U0001F251"
                           "]+", flags=re.UNICODE)
    return emoji_pattern.sub(r'', text)
```

```
def remove_url(text):
    url_pattern  = re.compile('http[s]?://(?:[a-zA-Z]|[0-9]|[$-
_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+')
    return url_pattern.sub(r'', text)
 # converting return value from list to string


def clean_text(text ):
    delete_dict = {sp_character: '' for sp_character in string.punctuation
}
    delete_dict[' '] = ' '
    table = str.maketrans(delete_dict)
    text1 = text.translate(table)
    #print('cleaned:'+text1)
    textArr= text1.split()
    text2 = ' '.join([w for w in textArr if ( not w.isdigit() and  ( not w
.isdigit() and len(w)>3))])

    return text2.lower()

# Connect to google drive and mount the directory
from google.colab import drive
drive.mount('/content/drive/')

# Load training data
train_data= pd.read_csv('/content/drive/My Drive/Colab Notebooks/Training
data.csv')
train_data.dropna(axis = 0, how ='any',inplace=True)
train_data['Num_words_text'] = train_data['Question'].apply(lambda x:len(s
tr(x).split()))
mask = train_data['Num_words_text'] >1
train_data = train_data[mask]
print('===========Train Data =========')
print(train_data['Class'].value_counts())
print(len(train_data))
print('===============================')

train_data['Question'] = train_data['Question'].apply(remove_emoji)
train_data['Question'] = train_data['Question'].apply(remove_url)
train_data['Question'] = train_data['Question'].apply(clean_text)

# Load testing data
test_data= pd.read_csv('/content/drive/My Drive/Colab Notebooks/Test data.
csv')
test_data.dropna(axis = 0, how ='any',inplace=True)
```

```
test_data['Num_words_text'] = test_data['Question'].apply(lambda x:len(str
(x).split()))
mask = test_data['Num_words_text'] >1
test_data = test_data[mask]
print('===========Test Data =========')
print(test_data['Class'].value_counts())
print(len(test_data))
print('==============================')

test_data['Question'] = test_data['Question'].apply(remove_emoji)
test_data['Question'] = test_data['Question'].apply(remove_url)
test_data['Question'] = test_data['Question'].apply(clean_text)

X_train, X_valid, y_train, y_valid = train_test_split(train_data['Question
'].tolist(), train_data['Class'].tolist(), test_size=0.25,stratify = train
_data['Class'].tolist(), random_state=0)


print('Train data len:'+str(len(X_train)))
print('Class distribution: '+str(Counter(y_train)))
print('Valid data len:'+str(len(X_valid)))
print('Class distribution: '+ str(Counter(y_valid)))

x_train=np.asarray(X_train)
x_valid = np.array(X_valid)
x_test =np.asarray(test_data['Question'].tolist())

le = LabelEncoder()

train_labels = le.fit_transform(y_train)
train_labels = np.asarray( tf.keras.utils.to_categorical(train_labels))

valid_labels = le.transform(y_valid)
valid_labels = np.asarray( tf.keras.utils.to_categorical(valid_labels))

test_labels = le.transform(test_data['Class'].tolist())
test_labels = np.asarray(tf.keras.utils.to_categorical(test_labels))
list(le.classes_)

train_ds = tf.data.Dataset.from_tensor_slices((x_train,train_labels))
valid_ds = tf.data.Dataset.from_tensor_slices((x_valid,valid_labels))
test_ds = tf.data.Dataset.from_tensor_slices((x_test,test_labels))
```

```
print(y_train[:10])
train_labels = le.fit_transform(y_train)
print('Text to number')
print(train_labels[:10])
train_labels = np.asarray( tf.keras.utils.to_categorical(train_labels))
print('Number to category')
print(train_labels[:10])

count =0
print('======Train dataset ====')
for value,label in train_ds:
    count += 1
    print(value,label)
    if count==5:
        break
count =0
print('======Validation dataset ====')
for value,label in valid_ds:
    count += 1
    print(value,label)
    if count==5:
        break
print('======Test dataset ====')
count=0
for value,label in test_ds:
    count += 1
    #if count<5:
    print(value,label)
    if count==5:
        break

embedding = "https://tfhub.dev/google/universal-sentence-encoder/4"
hub_layer = hub.KerasLayer(embedding, input_shape=[],
                           dtype=tf.string, trainable=True)
# 71 % val accuracy
model1 = tf.keras.Sequential()
model1.add(hub_layer)
model1.add(tf.keras.layers.Dense(256, activation='relu'))
model1.add(tf.keras.layers.Dense(144, activation='relu'))
model1.add(tf.keras.layers.Dense(72, activation='relu'))
model1.add(tf.keras.layers.Dense(36, activation='relu' ,kernel_regularizer
=regularizers.l1(0.1)))
model1.add(tf.keras.layers.Dense(6,activation='sigmoid' ,kernel_regularize
r=regularizers.l1(0.1)))
```

```
tf.keras.utils.plot_model(model1, "multi_input_and_output_model.png", show
_shapes=True)

epochs = 150

from datetime import datetime
from packaging import version
from tensorflow import keras
# Define the Keras TensorBoard callback.
logdir="logs/fit/" + datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = keras.callbacks.TensorBoard(log_dir=logdir)


history1 = model1.fit(train_ds.shuffle(100).batch(128),
                    epochs= epochs ,batch_size=32,
                    validation_data=valid_ds.batch(128),
                    callbacks=[tensorboard_callback],
                    verbose=1)

%tensorboard --logdir logs

plt.plot(history1.history['loss'], label='training data')
plt.plot(history1.history['val_loss'], label='validation data')
plt.title('Loss for Text Classification')
plt.ylabel('Loss value')
plt.xlabel('No. epoch')
plt.legend(loc="upper left")
plt.show()

plt.plot(history1.history['categorical_accuracy'], label='training data')
plt.plot(history1.history['val_categorical_accuracy'], label='validation d
ata')
plt.title('Categorical Accuracy for Text Classification')
plt.ylabel('Categorical Accuracy value')
plt.xlabel('No. epoch')
plt.legend(loc="upper left")
plt.show()

# Evaluate the model on the test data using `evaluate`
print("Evaluate on test data")
results = model1.evaluate(x_test,test_labels)
print("test loss, test acc:", results)
```

```
# Generate predictions (probabilities -- the output of the last layer)
# on test  data using `predict`
print("Generate predictions for all samples")
predictions = model1.predict(x_test)
print(predictions)
predict_results = predictions.argmax(axis=1)

test_data['pred_sentiment']= predict_results
test_data['pred_sentiment'] = np.where((test_data.pred_sentiment == 0),'An
alysis',test_data.pred_sentiment)
test_data['pred_sentiment'] = np.where((test_data.pred_sentiment == '1'),'
Application',test_data.pred_sentiment)
test_data['pred_sentiment'] = np.where((test_data.pred_sentiment == '2'),'
Comprehension',test_data.pred_sentiment)
test_data['pred_sentiment'] = np.where((test_data.pred_sentiment == '3'),'
Evaluation',test_data.pred_sentiment)
test_data['pred_sentiment'] = np.where((test_data.pred_sentiment == '4'),'
Knowledge',test_data.pred_sentiment)
test_data['pred_sentiment'] = np.where((test_data.pred_sentiment == '5'),'
Synthesis',test_data.pred_sentiment)

from sklearn.metrics import classification_report
labels = ['Knowledge', 'Comprehension','Application', 'Analysis', 'Evaluat
ion', 'Synthesis' ]

print(classification_report(test_data['Class'].tolist(),test_data['pred_se
ntiment'].tolist(),labels=labels))

model1.save('/content/drive/My Drive/Colab Notebooks/tf_model')
```

Appendix 2: Sample Code for Question classification API

```
from django.shortcuts import render

from django.http import Http404

from rest_framework.views import APIView

from rest_framework.response import Response

from rest_framework import status

import tensorflow as tf

from sklearn.metrics import classification_report

import numpy as np

from sklearn.preprocessing import LabelEncoder


class QuestionClass(APIView):

    new_model           =           tf.keras.models.load_model('C:/Users/hp
pavilion/Desktop/Apps/QCAPI/classify_question/tf_model')

    new_model.summary()

    def get(self, request, format=None):

        data ={'question':'Question', 'category':'Category'}

        return Response(data)

    def post(self, request, format=None):

        print("Question",request.POST['question'])

        question = request.POST['question']

        question_list = [question]

        x_test= np.asarray(question_list)

        predictions = self.new_model.predict(x_test)

        print(predictions)

        predict_results = predictions.argmax(axis=1)

        print("Predictions: ", predict_results)
```

```
        labels = ['Knowledge', 'Comprehension', 'Application', 'Analysis', 
'Evaluation', 'Synthesis']
        labels_cat=labels[predict_results[0]]
        print(labels_cat)
            data = {'question': question, 'category': labels_cat}
        return Response(data)
```

Appendix 3: Question classification Moodle LMS Plugin Code

```
<?php
require_once(__DIR__ . '/../../config.php');
global $DB, $CFG, $PAGE, $OUTPUT, $USER, $COURSE;
require_once($CFG->dirroot                                                   .
'/local/neuralnetworkquestionsclassifier/classes/form/question.php');
$base_url = $CFG->wwwroot;
$PAGE->set_url(new
moodle_url('/local/neuralnetworkquestionsclassifier/questions.php'));
$PAGE->set_context(\context_system::instance());
$PAGE->set_title('Classify questions');
$context = context_course::instance($_GET['course']);
$roles = get_user_roles($context, $USER->id, true);
$role = key($roles);
$rolename = $roles[$role]->shortname;
if(isset($_GET['course']) && $rolename!='student'  && $rolename!='guest' &&
$rolename!='teacher'){
    require_login();
    $course_id = $_GET['course'];
$lables   =   "('Knowledge',   'Comprehension',   'Application',   'Analysis',
'Evaluation', 'Synthesis')";
$questions = $DB->get_records_sql("SELECT q.id, q.questiontext,q.category,
qc.contextid, t.name
                                     FROM {question} q
                                     LEFT JOIN {question_categories} qc ON
qc.id = q.category
                                     LEFT   JOIN   {tag_instance}   ti   ON
ti.itemid = q.id
                                     LEFT JOIN {tag} t ON t.id = ti.tagid
                                     WHERE    t.name    is    null    and
q.category=$course_id or t.name not in $lables and ti.itemtype='question'
and q.category=$course_id ",[]);
$table = new html_table();
$table->head = array('SNo.','Question', 'Classification' , 'Context');
foreach ($questions as $question) {
    $questionname = (object)[
```

```
        'texttodisplay' => $question->questiontext,
    ];
    $id = $question->id;
    $postdata = http_build_query(
        array(
            'question' => strip_tags($question->questiontext)
        )
    );
    $opts = array('http' =>
        array(
            'method' => 'POST',
            'header' => 'Content-type: application/x-www-form-urlencoded',
            'content' => $postdata
        )
    );
    $context = stream_context_create($opts);
    // Set the API URL here
    $result = file_get_contents('http://127.0.0.1:8000/classify/', false,
$context);
    $res = json_decode($result);
    $q_text = $question->questiontext;
    $q_cat = $res->category;
    $context = context_system::instance();
    core_tag_tag::set_item_tags('core_question',
        'question', $question->id,
        $context,
        array($q_cat));
    $table->data[] = array($id, $q_text, $q_cat, $question->contextid); //
,$b,$c,$d,$e,'<a href="'.$link.'">View</a>'
}
echo $OUTPUT->header();
$templatecontext = (object)[
    'texttodisplay' => 'here is some text',
];
echo html_writer::table($table);
```

```
    echo                                    \core\notification::info("<a
href='$base_url/course/view.php?id=$course_id'>Go to back to course </a>");
echo $OUTPUT->footer();
}
else{
    echo $OUTPUT->header();
    echo  \core\notification::error("  You  arrived  here  by  mistake  <a
href='$base_url'>Take me to dashboard </a>");
    echo $OUTPUT->footer();
}
<?php
// This file is part of Moodle - http://moodle.org/
//
// Moodle is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// Moodle is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with Moodle.  If not, see <http://www.gnu.org/licenses/>.

/**
 * Version details.
 *
 * @package    local_neuralnetworkquestionsclassifier
 * @author     Peter Ndegwa
 * @copyright  Andreas Grabs
 * @license    http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or later
 */
function local_neuralnetworkquestionsclassifier_before_footer(){
    $message = 'My Notification Message'.$_GET['id'];
```

```
    $type = 'success';
    global $COURSE, $DB, $PAGE, $USER, $CFG;
    $url = $PAGE->url;
    $context = context_course::instance($COURSE->id);
    $roles = get_user_roles($context, $USER->id, true);
    $role = key($roles);
    $rolename = $roles[$role]->shortname;
    $course_id = $PAGE->course->id;
    $course = $DB->get_record('course', ['id' => $course_id]);
    if($course && $_GET['id']!="" && stristr($url, '/course/view.php?id=')
&& $rolename!='student'  && $rolename!='guest'){
        require_login();
        $lables    =    "('Knowledge',    'Comprehension',    'Application',
'Analysis', 'Evaluation', 'Synthesis')";
        $questions        =        $DB->get_records_sql("SELECT        q.id,
q.questiontext,q.category, qc.contextid, t.name
                                     FROM {question} q
                                     LEFT JOIN {question_categories} qc ON
qc.id = q.category
                                     LEFT   JOIN   {tag_instance}   ti   ON
ti.itemid = q.id
                                     LEFT JOIN {tag} t ON t.id = ti.tagid
                                     WHERE     t.name     is     null     and
q.category=$course_id or t.name not in $lables and ti.itemtype='question'
and q.category=$course_id ",[]);
        $base_url = $CFG->wwwroot;
        if(!empty($questions) && sizeof($questions)>0){
            \core\notification::info("You  have  question(s)  that  are  not
classified   by   Bloom's   Taxonomy   Model   for   $course->fullname.   <a
href='$base_url/local/neuralnetworkquestionsclassifier/questions.php?cours
e=$course_id'> CLICK THIS LINK TO CLASSIFY.</a>");
        }
    }
    function
local_neuralnetworkclassifiermenu_extend_navigation_frontpage(navigation_n
ode $frontpage) {
```

```
        $frontpage->add(
            get_string('pluginname',
'local_neuralnetworkquestionsclassifier'),
            new
moodle_url('/local/neuralnetworkquestionsclassifier/questions.php')
        );
    }
}
```

Appendix 4: Sample Labelled Questions

Knowledge

1. List two reference parameters in the setHour function
2. Explain briefly the meaning of the following terms: Demography, Consumer sales promotion , and Media mix
3. Label the parts of the diagram
4. Based on the above dataType class, list all the function members
5. Define morphology
6. Define Method in JAVA.
7. Define Inheritance concept
8. What is Encapsulation ?
9. Name the authors of...?
10. List the fractions you know and can show….
11. List the __________________ in descending order
12. List the attributes of the following shape.
13. List two local variables in the main function
14. List the data members
15. List important information you can remember from lesson….
16. Make a list of the main events of the story.
17. Memorize and recall the periodic table
18. Memorize the multiplication facts
19. Memorize the meaning of …
20. Name three 19th-century women English authors.
21. Name all the characters in the story.
22. Name five cities in US.
23. Recall the main components of the flowchart
24. Recall four facts from the story
25. What do you recall ...?
26. What is the definition of the following terms…

Comprehension

1. Can you explain what is happening . . . what is meant . . .?
2. Class Facility is an abstract class. Explain what this means
3. Draw a diagram explaining how air pressure affects the weather.

4. Describe how interest rates affect the economy
5. Explain what is happening in the first picture of the story.
6. Explain why the story has the title that it does.
7. Explain how the heart is like a pump.
8. Explain what is happening...
9. Explain what is meant.
10. Explain in one sentence what the method incFromN does
11. How would you explain...?
12. How would you express ________________?
13. How can you describe _____________?
14. How would you classify the type of ...?
15. How can you explain what is meant ...?
16. Draw a flowchart to accept a number and output its factorial.example 1: input 5 -> 5! = 5x4x3x2x1 -> output 120
17. Identify the key points in the text
18. Identify three mistakes from the passage and correct them.
19. State in your own words ...
20. State in one word ...
21. Put in your own words…
22. What is the difference between atomic and molecular mass?
23. In your own words, tell what the story is about.

Application

1. Design or sketch a marketing strategy for your product using a known strategy as a model.
2. Sketch an experiment to see how plants grow in different kinds of soil.
3. Sketch a diagram which shows these fractions or take photographs of the fractions.
4. Demonstrate and illustrate measures of central tendency and dispersion, write a journal entry.
5. Interpret the graph and state how many trees were cut down to produce paper.
6. How does the law of supply and demand explain the current increase in the price of fruit?
7. Predict what would happen if ...
8. Write a textbook about this topic for others.
9. Write what you might have done.
10. Write an explanation about this topic for others
11. Write a C++ statement to declare a variable of type musicType name MyTune.

12. Draw an illustration of the linked list after the execution of each of the statement below. (Use the original illustration for each question).
13. Write the definition of displaySize() method for each class
14. Suggest a way to deal with this scenario
15. Choose the best statements that apply ...

Analysis

1. Analyze the selected information.
2. Can you discriminate the difference parts . . . ?
3. Discriminate the pros and cons of ____.
4. Analyze the following questionnaire.
5. How can you categorize ____ according to...?
6. How can you distinguish the different parts of_____?
7. How can you differentiate between...?
8. Discriminate the different parts of the story (introduction, development, climax, resolution.)
9. Analyze a distinction. State the point of view of . . .
10. Categorize the story into parts and think of a good title for each of the parts.
11. Analyze a work of art in terms of form, color and texture.
12. Trace the contents of matrix from the following statements int matrix[3][2];int j, k;for (j = 0; j < 3; j++)for (k =h 0; k < 2; k++)matrik[j][k] = j + k;
13. Trace the value of alpha after the following code executes. int alpha[5];int j;alpha[0] = 5;for (j = 1; j < 5; j++){if(j % 2 == 0)alpha[j] = alpha[j – 1] + 2;else alpha[j] = alpha[j – 1] + 3;}
14. What distinctions can be made about...and...?
15. What is the relationship between probability and statistical analysis?
16. What is the analysis of _______________?
17. Draw a diagram to show the memory configuration after the following statements ....
18. What has the author used to create this effect
19. Compare and contrast the two items and produce a summary of their similarities and differences
20. Develop a set of criteria for distinguishing between a good and bad example of this.
21. What evidence can you find to suggest that people should be regarded as the most important resource of an organisation?
22. While the futures market is too risky for most investors, this risk creates the opportunities for large returns. Discuss.
23. What evidence can you list for...

Synthesis

1. Create several different strategies to solve a mathematical problem.
2. Can you propose an alternative plan to . . . ?
3. Can you develop method to handle the following case . . . ?
4. Can you propose a model that would change . . . ?
5. Can you think of an original way for collect and develop the . . . ?
6. Develop a menu for a new healthy foods restaurant
7. Design a scientific study to test the effect of different kinds of music on hens' egg production.
8. Design a building to house your study.
9. Design a record, book or magazine cover for...
10. Design a new monetary system or an experiment for establishing.
11. Create several scientific hypotheses to …
12. Propose an alternative way to solve …
13. Propose a set of alternatives for reducing dependence on fossil fuels that address both economic and environmental concerns.
14. Rewrite the story from an animal's point of view.
15. Rewrite the story briefly, but change someone or something in it. (For example, substitute a dog for a wolf in The Three Little Pigs).
16. Use your imagination to create a picture about the story. Then, add one new thing that was not in the story.
17. What hypotheses can you develop based on the data ? Why ?
18. Write a journal from the point of view of mountaineer.
19. Write about your feelings in relation to...
20. Write a TV show play, puppet show, role play, song or pantomime about..
21. Write a poem about the story.
22. Write 5 new titles for the story that would give a good idea of what it was about
23. Write another ending to the story that is different from the author's ending.
24. Write a JAVA program to show the Overloading concept
25. Write a program that uses nested loops to print the following output:
26. Write a program to calculate the SQRT for Array elements, you may use the Math class function (sqrt), the Array should include 5 Integer elements. The output should be the array elements along with their SQRT value.
27. Create an advertising campaign

28. Construct an alternative way to ...
29. Devise a different way of using X
30. Merge the different ideas into a single solution.

Evaluation

1. Can you defend your position about ... ?
2. Can you assess the value or importance of . . . ?
3. What limitation does X have?
4. Develop a proof ... and justify each step ...
5. "Don’t use public instance variables" is defensive programming techniques. discuss why it is good advice?
6. How would you justify . . . ?
7. Judge the value of... What do you think about...?
8. Judge how well a project meets the criteria of a rubric.
9. Judge the validity of arguments for and against astrology
10. Justify the concept of inheritance and give the sample of code to illustrate your answer.
11. Justify the object oriented programming concept
12. Prepare a case to present, and evaluate your view about ...
13. Prepare a list of criteria to judge ... Evaluate expressions
14. Rate the following recommendations and set the priority ______.
15. What data was used to evaluate _____________?
16. What judgments can you make about...?
17. What criteria would you use to assess...?
18. What criteria would you use to evaluate if your answer is correct?
19. What would you cite to defend the actions . . . ?
20. Outline how class ArrayList could be implemented using an array
21. Can you justify the decisions you have made ?
22. How would you rank the items given the criteria ?
23. Your advice has been sought to settle the following dispute in Company X. Referring to appropriate legal principles, write a short report advising the company on the best course of action to adopt.
24. What criticisms could you make ?