ENHANCED PARTS OF SPEECH (POS) WEIGHTED TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) IN QUESTION CLASSIFICATION OF EXAMINATION QUESTION BASED ON BLOOM'S TAXONOMY

MOHAMMED OSMAN GANI

MASTER OF SCIENCE (COMPUTER SCIENCE)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN
JANUARY 2024

ENHANCED PARTS OF SPEECH (POS) WEIGHTED TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) IN QUESTION CLASSIFICATION OF EXAMINATION QUESTION BASED ON BLOOM'S TAXONOMY

By

MOHAMMED OSMAN GANI

A dissertation submitted to the Department of Computer Science,
Faculty of Information and Communication Technology,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Science (Computer Science)
January 2024

DEDICATION

To my beloved parents,

Your unwavering support, endless encouragement, and boundless love have been the guiding stars throughout my academic journey. This thesis is a tribute to your sacrifices and the values you instilled in me. Thank you for believing in me and being my constant source of inspiration. This accomplishment is as much yours as it is mine.

ABSTRACT

ENHANCED PARTS OF SPEECH (POS) WEIGHTED TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) IN QUESTION CLASSIFICATION OF EXAMINATION QUESTION BASED ON BLOOM'S TAXONOMY

Mohammed Osman Gani

In contemporary educational settings, the traditional practice of utilising examinations as a means of assessing students' knowledge persists. The creation of a well-crafted question paper is considered an effective method for assessing students' understanding across various cognitive levels. The examination question classification (EQC) process plays a crucial role in achieving the goal of producing high-quality question papers for assessing students at different cognitive levels. EQC determines the cognitive levels of questions and assigns the cognitive level to questions using Bloom's Taxonomy (BT) cognitive domain. However, manually assigning cognitive levels is time-consuming, and not all educators possess a thorough understanding of the BT cognitive domain. As a result, the researchers focused on automating the EQC using machine learning (ML) and deep learning (DL) to overcome the aforementioned challenges. Numerous previous studies focused on enhancing the accuracy of the EQC based on BT by enhancing term weighting schemes. However, these studies assigned equal weight to two distinct categories of verbs in the questions: BT action verbs and supporting verbs. It is important to note that BT verbs possess a more significant influence in determining the cognitive level of a question than

supporting verbs. Consequently, the primary objective of this study is to introduce the ETFPOS-IDF term weighting model, which assigned a higher weight to BT action verbs than supporting verbs. In addition, the effectiveness of the supervised term weighting (STW) scheme, which has never been addressed before in EQC, was investigated. Furthermore, a comparison was performed between the proposed term weighting model and existing DL models proposed in past studies. This study used three classifiers: support vector machine, artificial neural network, and random forest, and five datasets, three of which were from past studies; one was newly collected, and the fifth was formed by merging the other four datasets. The accuracy and F1 score were utilised as evaluation metrics. The experimental results showed that the proposed term weighting model outperformed both existing term weighting schemes and DL models and achieved an accuracy of 82.8% and an F1 score of 82.9% during cross-validation and 87.1% in both metrics in the train-test split scenario. The outcomes of this study indicated that differentiating between different verb types significantly increases the classification accuracy of examination questions. Regarding STW schemes, this study found no superiority over unsupervised term weighting (USTW) schemes. Future work may involve identifying the optimal weight difference for verb types, hybridising STW and USTW schemes, and exploring the effectiveness of large language models.

ACKNOWLEDGEMENT

I would like to begin my acknowledgements by expressing my gratitude to Allah, the most merciful and compassionate, for granting me strength, wisdom, and perseverance throughout my journey in completing this thesis.

I am deeply thankful to my supervisor, Dr Ramesh Kumar Ayyasamy, for his unwavering support, valuable guidance, and unending patience. Your expertise, dedication, and mentorship have been instrumental in shaping this research.

I would also like to extend my heartfelt appreciation to my co-supervisors, Dr Anbuselvan Sangodiah and Mr Yong Tien Fui. Your insights, constructive feedback, and dedication to my academic growth have been invaluable, and I am truly grateful for your contributions to this work.

I am forever thankful to my family and friends, whose love and tireless belief in me have been my anchor throughout this journey. Your sacrifices and encouragement have propelled me forward and kept me motivated.

I extend my heartfelt thanks to everyone who has played a role in this thesis, no matter how big or small. This work would not have been possible without your support, and I am deeply grateful for your contributions to my academic and personal growth.

APPROVAL SHEET

This dissertation/thesis entitled **"ENHANCED PARTS OF SPEECH (POS) WEIGHTED TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) IN QUESTION CLASSIFICATION OF EXAMINATION QUESTION BASED ON BLOOM'S TAXONOMY"** was prepared by MOHAMMED OSMAN GANI and submitted as partial fulfillment of the requirements for the degree of Master of Science (Computer Science) at Universiti Tunku Abdul Rahman.

Approved by:

17-Jan-2024

Dr Ramesh Kumar Ayyasamy Date:……..
Main Supervisor
Faculty of Information and Communication Technology
Universiti Tunku Abdul Rahman, Kampar, Perak

17-Jan-2024

Mr Yong Tien Fui Date:……..
Co-Supervisor
Faculty of Information and Communication Technology
Universiti Tunku Abdul Rahman, Kampar, Perak

17-Jan-2024

Ts Dr Anbuselvan Sangodiah Date:……..
Co-Supervisor
Faculty of Computing and Engineering
Quest International University, Ipoh, Perak

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 17th January 2024

SUBMISSION OF DISSERTATION

It is hereby certified that **Mohammed Osman Gani** (ID No: **21ACM07042**) has completed this dissertation entitled "ENHANCED PARTS OF SPEECH (POS) WEIGHTED TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) IN QUESTION CLASSIFICATION OF EXAMINATION QUESTION BASED ON BLOOM'S TAXONOMY" under the supervision of **Dr Ramesh Kumar Ayyasamy** (Supervisor) from the Department of Information Systems, Faculty of Information and Communication Technology, **Mr Yong Tien Fui** (Co-Supervisor) from the Department of Information Systems, Faculty of Information and Communication Technology, and **Ts Dr Anbuselvan Sangodiah** (Co-Supervisor) from the Faculty of Computing and Engineering, Quest International University.

I understand that University will upload softcopy of my dissertation in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

mohammed osman gani

_____

(Mohammed Osman Gani)

DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations, which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Signature: mohammed osman gani

Name: Mohammed Osman Gani

Date: 17th January 2024

LIST OF PUBLICATIONS

A portion of the research presented within this thesis draws upon the contents of the subsequent publications:

- M. O. Gani, R. K. Ayyasamy, S. M. Alhashmi, A. Sangodiah, and Y. T. Fui, "ETFPOS-IDF : A Novel Term Weighting Scheme for Examination Question Classification Based on Bloom's Taxonomy," *IEEE Access*, vol. 10, no. November, pp. 132777–132785, 2022, doi: 10.1109/ACCESS.2022.3230592.

- M. O. Gani, R. K. Ayyasamy, T. Fui, and A. Sangodiah, "USTW Vs. STW: A Comparative Analysis for Exam Question Classification based on Bloom's Taxonomy," *Mendel*, vol. 28, no. 2, pp. 25–40, 2022, doi: 10.13164/mendel.2022.2.025.

- M. O. Gani, R. K. Ayyasamy, A. Sangodiah, and Y. T. Fui, "Bloom's Taxonomy-based exam question classification: The outcome of CNN and optimal pre-trained word embedding technique," Educ. Inf. Technol., pp. 1–22, 2023, doi: 10.1007/s10639-023-11842-1.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
BOW	Bag of Words
BT	Bloom's Taxonomy
CF-DF	Category Frequency-Document Frequency
CNN	Convolutional Neural Network
X^2	Chi-Square
DF	Document Frequency
DFS	Distinguishing Feature Selector
DNN	Dense Neural Network
DL	Deep Learning
DT	Decision Trees
EDM	Educational Data Mining
EQC	Examination Question Classification
GR	Gain Ratio
GRU	Gated Recurrent Unit
ICF	Inverse Category Frequency
ICSDF	Inverse Class Space Density Frequency
IDFEC	Inverse Document Frequency Excluding Category
IF	Impact Factor
IG	Information Gain
IGM	Inverse Gravity Moment
IQF	Inverse Question Frequency
ITE	Inverse Term Entropy

KNN	K-Nearest Neighbour
LSTM	Long Short-Term Memory
LR	Logistic Regression
MI	Mutual Information
ML	Machine Learning
NB	Naïve Bayes
NLP	Natural Language Processing
OR	Odds Ratio
POS	Parts of Speech
QA	Question-Answering
QC	Question Classification
QF	Question Frequency
RNNs	Recurrent Neural Networks
SMEs	Subject Matter Experts
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency

CHAPTER 1

INTRODUCTION

1.1 Background

Question classification (QC) basically labels the question based on some pre-determined category. QC is involved in question-answering (QA) systems [1], dialogue systems [2], and many more natural language processing (NLP) tasks. The task of automatically generating responses to questions written in natural language is called the QA system. In the QA system, the process of QC categorises questions based on the type of answer they require or the topic they pertain to, which helps in identifying the most relevant documents to provide an answer [3]. Apart from the areas mentioned above, QC is involved in the educational context to label the examination questions based on their cognitive level [4], [5]. Therefore, the classification of examination questions falls within the purview of educational data mining (EDM), to be precise, within the domain of text mining application of EDM, given that examination questions constitute textual data. Examination question classification (EQC) is a supervised learning process, as it relies on pre-determined categories and is a form of text classification.

The classification of examination questions is the process of assigning labels to the questions based on their cognitive level using a framework such as Bloom's Taxonomy (BT) cognitive domain. This cognitive domain includes several levels, ranging from the simplest to the most complex. The purpose of

using such a framework is to create a set of questions that effectively evaluate the student's understanding of the course material in academic institutions. Traditionally, academicians have been responsible for manually labelling questions in academic institutions. However, this method of manual EQC is prone to several issues.

1. Not all evaluators or academicians are able to correctly label the questions [6].
2. It requires a significant amount of time to label the examination questions manually [7].
3. The classification process can vary due to differences in perception among different examiners [8].

Apart from the abovementioned issues, questions involving multiple cognitive skills may be challenging to categorise accurately using a single level from BT without universally standardised guidelines for applying BT. Moreover, the conventional manual labelling approach is typically static, making it difficult to adapt to individual student needs or learning styles.

The abovementioned issues can be overcome with the automation of EQC using data mining, more specifically through text mining technologies. The classification of examination questions has been a subject of much research in recent years, with many researchers utilising text-mining techniques to automate the process. Despite this, ongoing research remains needed to optimise and improve the classification process of examination questions.

To achieve better accuracy for EQC based on BT, previous researchers worked on feature selection [9], [10], feature set extraction [11], [12], and term weighting [7], [13]. Term weighting is a method of indicating the importance and significance of a term in a document by assigning a numerical weight to it. Despite previous attempts to enhance term weighting for BT-based QC, there are still limitations, such as treating all verbs in a question as having equal significance. Hence, there is potential to increase classification accuracy by enhancing term weighting. Therefore, this research focuses on enhancement in term weighting for BT-based QC to ensure the proper weightage for the term is applied.

1.2 Problem Statements

1.2.1 Lack of Consideration of Class Distribution in Existing Term Weightings

In NLP, two distinct approaches exist for term weighting: supervised and unsupervised approaches. The unsupervised term weighting (USTW) approach does not incorporate prior knowledge about the class distribution in the documents [14]; instead, it relies on the distribution of terms to assign weight. Conversely, the supervised term weighting (STW) approach considers the distribution of classes while determining term weight. Previous research [12], [13] in the area of EQC has not adequately considered class distribution in term weighting schemes, and as such, the STW approach has not been utilised. However, the STW approach helps to identify the terms that are most important for distinguishing between different classes. In the case of EQC, it is observed that certain terms recurrently appear in numerous questions and are associated

with the same cognitive level. Therefore, these terms exhibit a strong correlation with that particular cognitive level. Consequently, it is imperative to assign a greater weight to that particular cognitive level in the context of these terms as compared to other cognitive levels. Ignoring this aspect while weighting the terms may result in a certain percentage of questions being misclassified. USTW schemes, on the other hand, may assign high weights to terms that are common to all cognitive levels or to terms that are not very discriminative. Given this, this study investigates the potential effectiveness of STW schemes in improving classification accuracy for EQC in accordance with BT.

1.2.2 Lack of Proper Weightage for Verbs

Previous research has focused on improving the accuracy of classifying examination questions by enhancing the technique of term weighting. This technique involves assigning numerical values to terms to identify their importance level in the classification process. It is important to note that not all words within a question carry equal weight, as certain words have a greater impact than others. Additionally, it has been found that verbs are the most crucial parts of speech (POS) in determining the cognitive level of questions [12], [13]. However, two types of verbs may appear in a question: BT action verbs and supporting verbs. Previous research [7], [13] has assigned equal weight to both types of verbs, but action verbs should be given greater weight as they have a greater impact on determining cognitive levels. If both types of verbs are given equal weight, it might reduce the discrimination power during classification. As a consequence, this could elevate the likelihood of misclassification. Hence, this study addresses solving this issue.

1.3 Research Questions

RQ1: Does the STW scheme outperform the USTW scheme for EQC?

This research question aims to answer whether applying the STW scheme in the context of EQC tasks based on BT leads to better classification accuracy when compared to the USTW scheme.

RQ2: Does discriminating between the verb types while term weighting enhance EQC accuracy?

This research question centres on the discrimination between different types of verbs, focusing on BT action verbs receiving higher weight compared to supporting verbs. In essence, it answers the impact of verb type discrimination while term weighting on the performance of EQC.

RQ3: Does the proposed term weighting model outperform deep learning (DL) models for EQC based on BT?

This research question aims to answer whether the proposed term weighting model can compete with or potentially outperform state-of-the-art DL models when confronted with scenarios where the amount of training data is constrained.

1.4 Objectives of the Study

RQ1 → RO1: To investigate the effectiveness of the STW scheme for EQC based on BT.

This objective aims to compare the baseline STW scheme and the USTW scheme for EQC. It seeks to evaluate the effectiveness of the STW approach in

answering the research question associated with this objective, which may guide future research directions.

RQ2 → RO2: To propose an enhanced term weighting model to improve the accuracy of the EQC.

This objective seeks to propose an enhanced term weighting model where the discrimination between verb types is considered. The performance analysis and evaluation of the proposed term weighting model answered the research question of whether the discrimination between verb types enhances the accuracy of EQC.

RQ3 → RO3: To compare the proposed enhanced term weighting model with the DL models in the context of limited data.

DL models typically require a large amount of data to train. This objective intends to evaluate the efficacy of the newly proposed term weighting model compared to state-of-the-art DL models when dealing with scenarios where the amount of available training data is restricted.

1.5 Scope of the Research

This study focuses on classifying open-ended questions and does not cover close-ended questions. In open-ended questions, opinions and thoughts can be shared in-depth, and respondents are not compelled to select answers from a list. On the other hand, closed-ended questions, such as dichotomous and multiple-choice questions, present respondents with a list of possibilities from which to choose. However, questions from various domains were covered, such as Programming, Science, Social Science, Computing, Multimedia,

Mathematics, Business, and many more. Additionally, this study does not address coding-related programming questions requiring answers written in a programming language. Nonetheless, open-ended theoretical programming questions are covered in this research. The exclusion of close-ended and coding-related programming questions is motivated by their distinct format from open-ended questions. This study focused on POS-based weighting, especially distinguishing BT action and supporting verbs. However, close-ended and coding-related questions might not contain any BT action verbs. Hence, developing and evaluating the model with mixed questions from the abovementioned categories might lead to wrong assumptions about the methodology and technique.

1.6 Significance of the Study

In academic institutions, deploying an automated EQC model can benefit examination question setters. By using the model, academics can rapidly and effectively classify questions based on BT cognitive levels since automation eliminates the need for manual labour, reducing the potential for human error and allowing for faster and more consistent results. In addition to saving time and effort, this reduces the likelihood of misclassification by academicians. This is because not all academicians are competent in the BT cognitive domain. Furthermore, automation can also help create targeted assessments, leading to better evaluation of student's knowledge and progress. In addition, this technology can enhance the overall quality of examination question setting and evaluation in academic institutions.

1.7 Thesis Structure

This dissertation is composed of five chapters. An overall overview of each chapter is as follows:

CHAPTER 2: RELATED WORK – This chapter discussed the EDM and BT cognitive domain and reviewed previous research on classifying examination questions based on BT and STW in text classification. Additionally, the research gap identified after reviewing existing literature is discussed in this chapter.

CHAPTER 3: METHODOLOGY – This chapter details the research phases and the methods to achieve the objectives of this research. The approaches utilised to collect data, train and evaluate the proposed term weighting model, comparison of STW and USTW schemes, and DL models are discussed in this chapter.

CHAPTER 4: RESULTS AND DISCUSSION – This chapter reports the results of the proposed question classification model, a comparison of STW and USTW schemes, and the comparison of the proposed model with the existing DL models proposed in past literature. Furthermore, this chapter explains the significance of these findings and establishes a cohesive connection with the research questions formulated earlier.

CHAPTER 5: CONCLUSION – This chapter gives an overview of the dissertation, highlighting the research objectives, methodology and key findings. In addition, this chapter also reported the contributions and limitations of this research and suggested potential future research.

CHAPTER 2

RELATED WORK

This chapter thoroughly reviewed the past work on EQC based on BT. This study compared existing term weighting schemes of EQC with the STW of text classification. Hence, this section comprehensively discussed the STW schemes in text classification. Besides, EDM and its application are discussed, as well as BT.

2.1 Educational Data Mining

The principal objective of data mining techniques is to discover and extract patterns from stored data using various methods and algorithms [15]. When using data mining techniques, knowledge extraction is not confined to one type of database. Knowledge from several types of databases, such as transactional, active, object-oriented, spatial, and relational databases, can be extracted using data mining techniques [16]. Data collection is no longer a challenge; extracting valuable insights and presenting them to the user is a real challenge [17]. Various data mining methods, such as classification, generalisation, clustering, characterisation, pattern matching, and association, have recently been developed [16], [18]. The applicability of the data mining techniques is not limited to one sector or industry. Healthcare, education, business, social sciences, engineering, and economics are a few industries that use data mining [8]. An example of a data mining application is recommendation systems [19] in social networks. The recommendation systems in social

networks recommend new contact [17] to the user by analysing social networking data. Among the sectors where data mining applies, EDM [20], [21] is one of them.

The EDM community website [22] defines EDM as follows, *"Educational Data Mining is an emerging discipline, concerned with developing methods for exploring the unique and increasingly large-scale data that come from educational settings and using those methods to better understand students, and the settings which they learn in."* According to [23], *"The EDM process converts raw data coming from educational systems into useful information that could potentially have a great impact on educational research and practice."* Educational data includes the enrollment process of students and grades [24], administrative data from schools and universities [22], web data such as student-computer interaction [25], questionnaires [23], course information [26], and many more. EDM uses data mining techniques such as classification, clustering, association-rule mining, sequential mining, text mining, and many more [23], [26].

There are many applications in EDM, such as computer-supported collaborative learning [25], predicting student performance [27], student behaviour prediction [28], detecting undesirable student behaviours [29], classification of examination questions [30], and more. Predicting student performance is an application that uses data mining techniques and approaches to predict students' future performance based on their prior records. Bayesian classification, decision tree (DT), neural networks, rule-based methods, and feature selection are the most used techniques for predicting student performance

[29]. By analysing institution-generated data, several data mining models might be used to assist educational decision-making, thus leading to more impactful learning and the standard of education [31]. So, currently, EDM is drawing much attention, making it a new and rapidly growing research community [26].

2.2 Examination Question Classification and Bloom's Taxonomy

The written examination is the most conventional and traditional method of evaluating students in educational institutions [7]. The purpose of teaching and learning can be attained through cognitive evaluations [9]. In institutions, written examinations measure students' acquired ability, knowledge, and skills in contrast to the objectives and aims of the courses delivered over the semester. [32]. Questioning is widely recognised as an effective educational approach; nearly 80% of all teacher-student interactions are based on examinations [33].

An improperly prepared assessment may not accurately evaluate students' abilities, thereby impacting their scores and development through their academic program [34]. When creating assessments, it is essential to verify that each test or assignment given to students corresponds to the course learning goals [35]. However, creating assessments is a challenging task for evaluators, particularly when attempting to produce questions that cover a variety of cognitive levels to provide a fair and high-quality set of questions [9]. As a result, many evaluators are attempting to use a framework such as BT when creating examination questions to ensure that high-quality examinations are produced [7].

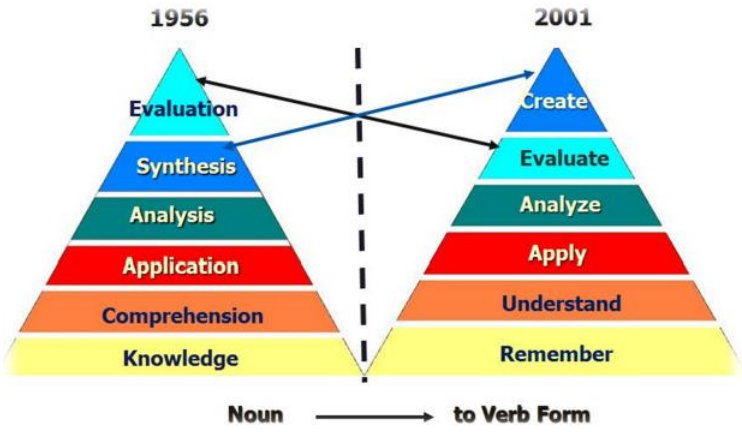

Figure 2.1: Cognitive domain of original and revised Bloom's Taxonomy [36]

BT is a fundamental concept that aids educators in formulating learning objectives, designing the curriculum, and generating assessments in the field of education [37]. An American educational psychologist, Benjamin Samuel Bloom, and his collaborators proposed BT [38] in 1956. The BT proposed by Benjamin Bloom consists of three domains: cognitive, emotional, and psychomotor [39]. In 2001, Anderson and Krathwohl [40] published a revised version of the cognitive domain of BT. In the revised version, Anderson and Krathwohl changed the names of levels and replaced nouns with verbs (Figure 2.1). The cognitive domain of BT has been widely used in past studies to classify examination questions [41], [42] and design learning objectives [5], [43]. The definition of each cognitive level is given below from the book [38], in which the BT cognitive domain was proposed, and sample action verbs from an earlier study [13] of EQC.

Knowledge

Definition: *"Involves the recall of specifics and universals, the recall of methods and processes, or the recall of a pattern, structure, or setting."*

Action verbs: Define, Name, Label, Order, Recall, List.

Comprehension

Definition: *"Refers to a type of understanding or apprehension such that the individual knows what is being communicated and can make use of the material or idea being communicated without necessarily relating it to other material or seeing its fullest implications."*

Action verbs: Clarify, Classify, Identify, Interpret, Illustrate.

Application

Definition: *"The Use of abstractions in particular and concrete situations. The abstractions may be in the form of general ideas, rules of procedures, or generalised methods, The abstractions may also be technical principles, ideas, and theories which must be remembered and applied."*

Action verbs: Apply, Assess, Develop, Prepare.

Analysis

Definition: *"The breakdown of a communication into its constituent elements or parts such that the relative hierarchy of ideas is made clear and/or the relations between ideas expressed are made explicit."*

Action verbs: Compare, Distinguish, Contrast, Arrange.

Synthesis

Definition: *"The putting together of elements and parts so as to form a whole. This involves the process of working with pieces, parts, elements, etc. and arranging and combining them in such a way as to constitute a pattern or structure not clearly there before."*

Action verbs: Compile, Create, Design, Generate.

Evaluation

Definition: *"judgments about the value of material and methods for given purposes. Quantitative and qualitative judgments about the extent to which material and methods satisfy criteria."*

Action verbs: Justify, Evaluate, Judge, Predict, Defend, Decide.

The classification of examination questions into BT cognitive level can be framed as a text classification task [10]. However, unlike normal texts, a question generally consists of a brief sentence, and most terms occur just once [44], which makes QC a unique challenge [12]. According to past studies, there are three approaches to classify examination questions based on BT's cognitive domain: machine learning (ML)-based [33], [45], rule-based [46], [47], and DL-based [48], [49]. Classification of the examination questions is not limited to one field or domain. Programming questions [9], [32], computer science questions [50], mathematics [45], science [12], computing [12], business[12], social [45], multimedia [12], and many more were used in past studies. The related work in EQC in accordance with BT involved feature selection, feature extraction, and

DL, which are discussed in the next subsections. The Feature extraction is subdivided further into feature set and term weighting.

2.3 Feature Selection in Examination Question Classification

Table 2.1 shows the feature selection techniques utilised in EQC based on BT. Multiple feature reduction techniques, document frequency (DF) and category frequency-document frequency (CF-DF) were analysed by Yusof and Hui [6] in order to decrease the feature space for the artificial neural network (ANN). They collected a dataset consisting of 274 questions and labelled the questions by the subject matter experts (SMEs). Their experiment demonstrated that DF is a suitable technique for reducing the number of features for EQC since it reduces the convergence time despite maintaining precision. The results showed that the lowest convergence error was 0.00092, and the highest classification precision was 65.26% without applying any feature reduction technique. The findings further showed that the whole feature set outperformed DF and CF-DF, and CF-DF was found to be inappropriate since it excluded BT verbs that might exist at several cognitive levels.

Table 2.1: Feature selection in examination question classification

Research Work	Year	Method	BT Version
[6]	2010	DF, CF-DF	Original
[10]	2012	TF	Original
[9]	2015	X^2, MI, OR	Original

Yahya et al. [10] investigated the effectiveness and introduced term frequency (TF) as a feature selection method for support vector machine (SVM) in classifying item bank questions into BT cognitive levels. Additionally, they investigated the performance of SVM with or without stop word removal. Before applying the SVM, the text was preprocessed by tokenising, punctuation removing, stemming, term weighting, and length normalisation. The performance of SVM was evaluated by considering the effect of TF and stop word removal. The result showed that removing stop words does not significantly improve SVM's performance. Their experiment result showed an average accuracy of 0.923 and micro F1 of 0.717 when TF >=2.

To improve the accuracy of the QC based on BT, a study [9] proposed an approach where they tested several feature selection methods and classifiers. SVM, naïve Bayes (NB), and k-nearest neighbour (KNN) were used to classify the questions. They tested these classifiers with the feature selection methods such as chi-square (X^2), mutual information (MI), and odds ratio (OR). This study introduced the majority voting technique in EQC and used it for the final classification to reduce misclassification. The voting algorithm delivered the best performance (macro F1 = 92.28) when MI was applied with a weighted feature size of 250. Despite having a higher overall macro F1 score of 92.28, the classification accuracy of cognitive levels such as Evaluation and Analysis is much lower than the other levels.

2.4 Feature Set in Examination Question Classification

Table 2.2 shows the feature set introduced in EQC based on BT. Sangodiah et al. [51] highlighted that most past studies in EQC focused on the bag of words (BOW) and syntactic features. According to them, an improvement in extracting the feature set is necessary. Hence, they introduced several feature sets, including keywords of the questions, headwords, and semantic and syntactic features. They suggested using the SVM for classification because of its high accuracy. However, no investigation was performed on whether the proposed feature sets improved the accuracy of EQC based on BT.

For decent accuracy, term frequency-inverse document frequency (TF-IDF) or N-gram requires a large amount of data, according to Kusuma et al. [45]. So, they introduced an approach to classify Indonesian language examination questions with lexical and syntactical features. Lexical features included N-gram, question length, word shape, and WH-word, whereas syntactical features were tag unigram, POS tagging, and headword. Their experiment results showed an accuracy of 94% with the linear kernel of SVM.

Osman and Yahya [11] investigated whether the combination of linguistically motivated features with several classifiers can increase QC accuracy or not. The authors tested and compared different ML classifiers, NB, SVM, logistic regression (LR), and DT, to classify examination questions based on BT. N-grams, the BOW, and POS were the linguistically motivated features used in classification. This study contributed by extracting and introducing the combination of linguistically motivated features in EQC. Their experiment

results showed that the highest accuracy achieved by SVM was 0.7667 with the single feature unigram. The highest accuracy (0.7683) was achieved by LR while combining two feature sets, such as unigrams and bigrams. The outcome of this study showed that the combination of linguistically motivated features might increase the accuracy of the EQC based on BT.

Existing feature types in the prior studies, as noted by Sangodiah et al. [12], may perform relatively well only on data sets containing questions that are too specific to one field or area. As a result, they introduced a new taxonomy-based feature to enhance the performance of BT-based EQC for datasets with questions from several fields. They tested the SVM with and without taxonomy-based features to investigate whether taxonomy-based features enhance classification accuracy or not. Their experiment result showed that the usage of taxonomy-based features increased the accuracy. The combination of BOW, POS, and taxonomy-based features achieved an accuracy of 0.729 in the dataset containing questions from multiple domains or areas and 0.754 in the dataset containing questions from one area or domain. This study contributed by introducing new feature sets called taxonomy features.

A study was conducted by Aninditya et al. [8] to achieve better classification accuracy by testing and comparing multiple feature sets for EQC based on BT with NB. They extracted words, N-Gram, and characters and tested these features with TF-IDF. This study labelled the six levels of the BT cognitive domain into high-order and low-order for classification, making it a binary classification problem. The first three levels were labelled as low-order and the rest of the three as high. Their experiment results showed the highest precision

of 85% with the N-Gram TF-IDF and recall of 82% with the words TF-IDF. This study contributed by extracting the abovementioned new feature sets.

Table 2.2: Feature set introduced by past studies in examination question classification

Research Work	Year	Features	BT Version
[51]	2014	Keyword, headword, and semantic features	Original
[45]	2015	Lexical and Syntactical features	Revised
[11]	2016	Combination of N-grams, the BOW, and POS	Original
[12]	2017	Taxonomy features	Original
[8]	2019	Words N-Gram and Characters	Revised

2.5 Term Weighting in Examination Question Classification

Omar et al. [52] proposed a rule-based approach to classifying the examination questions based on the cognitive domain of BT. They highlighted that the BT levels contain overlapping keywords. If only the questions' keywords are considered while classification, a question may belong to more than one category. They proposed category weighting, as illustrated in Table 2.3, to address the issue of keyword overlapping, in which conflicting categories were assigned weights. The weight was calculated based on the question's category from SMEs. The contribution of this study is the introduction of category weighting to solve BT keyword overlapping issues.

Another rule-based study [53] used the highest path and lemma similarities to classify the examination questions. They first obtained each question's verbs through POS tagging. After that, the path similarity for those

verbs was obtained and summed. The category with the highest path similarity value was determined as the category of that question. If the total path similarity value of the question is the same for multiple categories, the weight was assigned based on lemma similarity to determine the final category of the question. Their experiment results showed that among the 62 questions, 51 questions were classified correctly with the generated ruleset. This study contributed by introducing the highest path and lemma similarities in EQC.

Jayakodi et al. [54] proposed an approach to generate an appropriate ruleset using NLP, the WordNet similarity algorithm, and cosine similarity. They assigned the weight to the questions based on WordNet and Cosine similarity values to each question category. The result showed that the generated ruleset correctly classified 32 questions out of 45. They also compared several WordNet similarity algorithms to identify the most appropriate algorithm. The result showed Path similarity with 84% accuracy as the most optimal one in the context of the EQC. Proposing WordNet and cosine similarity approach, as well as identifying the best WordNet similarity algorithm for EQC, are the contributions of this study.

The unsupervised schemes TF-IDF [8], [11], [33], and binary [12] were also used in a few other studies of EQC. In both studies [11], [33], the authors applied the length normalisation after calculating term weighting. According to Sangodiah et al. [12], TF and TF-IDF perform effectively when the corpus's words or terms are repetitive. However, this is not the case with EQC, as questions usually contain fewer words. Therefore, Sangodiah et al. [12] utilised binary term weighting rather than TF and TF-IDF.

Table 2.3: Term weighting scheme in examination question classification

Research Work	Year	Scheme	Approach	BT Version
[52]	2012	Category weighting	Rule-based	Original
[53]	2015	Category weighting	Rule-based	Revised
[54]	2016	Category weighting	Rule-based	Revised
[11]	2016	TF-IDF	ML-based	Original
[12]	2017	Binary	ML-based	Original
[7]	2018	ETF-IDF	ML-based	Original
[13]	2020	TFPOS-IDF	ML-based	Original
[55]	2021	TF, TF-IDF	ML-based	Original

According to Mohammed and Omar [7], verbs play an important role in determining a question's cognitive level, meaning that verbs have a higher impact than other POS. On the other hand, nouns and adjectives are more crucial than other POS. So, they introduced an enhanced TF-IDF weighting scheme to classify the questions based on the cognitive domain of BT. The impact factor for the words was assigned with the help of the POS tagger. Among the POS, the highest impact factor was assigned to the verbs, after adjectives and nouns, then the remaining POS. Several classifiers were used in this study for the experiment, such as SVM, NB, and KNN. The proposed method of this study showed improvement in EQC. Their experimental result showed that SVM (86%) outperformed other classifiers, such as NB (85%) and KNN (81.6%), in terms of weighted F1-measure. This study contributed by introducing POS-based weighting in EQC.

Mohammed and Omar conducted another study in later years [13]. In this study, they proposed another term weighting scheme, TFPOS-IDF. Apart from term weighting, they tried to give a solution for the keyword overlapping issue of cognitive levels. The enhanced term weighting scheme TFPOS-IDF was combined with the word embedding technique word2vec as a solution for keyword overlapping. For the experiment, several classifiers, such as KNN, LR, and SVM, were used. The proposed method was tested with multiple datasets. SVM performed better in both datasets and achieved 83.7% and 89.7% in weighted F1 measures. Their experimental result showed that combining the enhanced TFPOS-IDF and Word2Vec improved classification accuracy. However, no investigation was conducted to evaluate how far the proposed method can solve the overlapping keyword problem. The study contributed by proposing an enhanced term weighting scheme TFPOS-IDF and introducing the pre-trained word embedding Word2Vec in ML-based EQC.

Sangodiah et al. [55] conducted a study to identify the optimal variant of TF and TF-IDF in the context of EQC based on BT since there are many variants of these schemes in text classification and EQC. They used SVM and NB to train the models, where accuracy and F1 score were used to evaluate those models. Their experiment results showed that the normalised TF-IDF outperformed all the TF and TF-IDF variants.

2.6 Deep Learning in Examination Question Classification

A few research have been conducted on EQC with DL, as shown in Table 2.4. In 2020, Das et al. [48] proposed DL models in classifying examination questions, with 3000+ questions in the training set. Their study was the first attempt to apply DL in EQC based on BT. The authors ensembled questions from past studies as well as some of the questions they collected and annotated by the experts. For the classification, they fine-tuned a pre-trained BERT model. With the BERT model, they achieved an overall accuracy of 89.67% and 88.68% for the WH questions. However, this study used accuracy as an evaluation metric despite the dataset not being balanced, as some classes have fewer questions than others.

Table 2.4: Work on deep learning in examination question classification

Research Work	Year	Algorithm	BT version
[48]	2020	BERT	Revised
[49]	2021	BERT	Original
[56]	2021	CNN, LSTM	Revised
[5]	2021	LSTM + FastText	Revised
[4]	2022	BERT + DNN	Original
[57]	2023	IndoBERT	Original

Another BERT-based DL study [49] was conducted to classify programming questions. They used a dataset of 504 questions, which experts labelled according to the BT cognitive domain. Among the 504 questions, only four questions belong to the evaluation level, and 20 and 17 questions belong to the synthesis and application levels, respectively. The rest of the questions

belong to the other three levels, which made the dataset extensively imbalanced. The authors conducted three different experiments with different class numbers. They utilised all six classes in the first trial and attained 59.2% accuracy. The accuracy (68.52%) improved in the second experiment, where the Evaluation, Synthesis, and Application classes were combined into a single class. The last experiment removed the previously mentioned three classes from the datasets and trained and evaluated the model with the rest of the low-order cognitive levels. The accuracy obtained in the final experiment was 82.61%.

Laddha et al. [56] compared the performance of long short-term memory (LSTM) and convolutional neural network (CNN) to classify the software engineering course questions. Their dataset consisted of 844 questions, with 70% data used in training and the rest of the 30% in testing. The first two cognitive levels consisted of most questions (666), which imbalanced the dataset. Their experiment results showed that the CNN achieved 80% accuracy and outperformed LSTM by 9%.

Shaikh et al. [5] compared three famous word embedding techniques: Word2Vec, GloVe, and FastText. No prior research before this has examined the word embedding approaches for EQC. FastText was found to be the optimal embedding technique. Therefore, the authors proposed a DL model with LSTM and FastText. They trained the model using the Yahya et al. [10] dataset. The dataset was initially labelled with the original BT and consisted of 600 questions. However, they used revised BT in their study. This study used 95% of data in training and 5% in testing. Consequently, only 30 questions were used in the testing. Their experiment results showed that the proposed LSTM achieved 87%

accuracy and 82% in F1 score with the train-test split and accuracy of 81% with the 10-fold cross-validation.

Sharma et al. [4] presented a novel pipeline for generating and classifying questions within the cognitive domain defined by BT. To achieve this, they construct a model trained on labelled datasets, enabling the categorisation of generated questions based on their cognitive level. The authors meticulously evaluate the performance of diverse DL architectures, including ConvNet1D, LSTM, Bidirectional LSTM, and BERT. Furthermore, they investigate the efficacy of various embedding techniques, encompassing the TensorFlow embedding layer, ELMo, and GloVe. The experimental results reveal that when coupled with a dense neural network (DNN), the BERT model performs better than the other architectures and embedding methods. Notably, the BERT model achieves an impressive accuracy of 0.811 and an F1 score of 0.810, solidifying its position as the most effective approach for the proposed task.

Baharuddin and Naufal [57] presented a novel classification system utilising the IndoBERT pre-trained model to categorise Indonesian multiple-choice examination questions according to BT. Their methodology employed hyperparameter fine-tuning to optimise model performance, which was subsequently evaluated through metrics such as accuracy, F1 score, precision, recall, and training/validation time. The findings suggested the system’s potential as a reliable tool for classifying Indonesian elementary school examination questions according to Bloom's cognitive levels.

2.7 Supervised Term Weighting in Text Classification

Statistics-based feature selection techniques: TF- information gain (IG), TF- X^2, and TF- gain ratio (GR) were proposed as weighting schemes by Debole and Sebastiani [58], as illustrated in Table 2.5. Their experiment result demonstrated that the TF-GR outperformed the other two schemes. They compared the performance of these STW schemes with the TF-IDF and did not consistently outperform the TF-IDF. These schemes were devised mainly for binary classification [59] and are computationally expensive for multi-class text classification.

Table 2.5: Supervised scheme in Text Classification

Research Work	Year	Scheme
[58]	2003	TF-IG, TF-GR, TF- X^2
[60]	2011	QF*ICF, IQF*QF*ICF
[61]	2013	TF-ICF
[62]	2013	TF-IDF-ICF, TF-IDF-ICSDF
[63]	2015	TF-IDFEC, TF-IDFEC-based
[64]	2016	TF-IGM, RTF-IGM
[59]	2017	TF-ITE, TP-ITE, ATF- ITE
[65]	2019	TF-IGM_{imp}, $RTFIGM_{imp}$
[66]	2021	TF-DFS, TF-MDFS

Repetition of the terms in questions is rare, so the TF of the terms is usually 1. According to Quan et al. [60], it is difficult to claim that terms with higher occurrences have higher significance than those with relatively lower occurrences. Hence, instead of using TF, they used question frequency (QF). By combining QF and inverse question frequency (IQF), they proposed two

schemes: QF* inverse category frequency(ICF) and IQF*QF*ICF. These two schemes can be used for both binary and multi-class classification processes.

Wang and Zhang [61] proposed TF-ICF in 2013, which differs from the QF*ICF since it uses TF instead of QF. The authors compared the performance of the TF-ICF with USTW schemes TF and TF-IDF. Their experiment results showed that the TF-ICF consistently outperformed the aforementioned USTW schemes. Ren and Sohrab [62] introduced a new STW scheme TF-IDF-ICF by combining IDF and TF-ICF. According to them, TF-ICF and TF-IDF favour rare terms while weighting the terms. So, they replaced the ICF of TF-IDF-ICF with inverse class space density frequency (ICSDF) and proposed another scheme, TF-IDF-ICSDF. Their experiment results showed that the TF-IDF-ICSDF creates positive discrimination in rarely and frequently occurred terms.

Domeniconi et al. [63] devised two supervised variations of TF-IDF in which the IDF part was replaced by inverse document frequency excluding category (IDFEC) and IDFEC-based. Chen et al. [64] proposed a new STW scheme known as TF- inverse gravity moment (IGM) and a variant known as RTF-IGM, where RTF is the square root of TF. They compared the performance of the proposed schemes to the performance of four different STW schemes: TF-X^2, TF-Prob, TF-IDF-ICSDF, and TF-RF. The findings of their experiment demonstrated that the TF-IGM and RTF-IGM outperformed these four STW schemes. Both research mentioned above [63], [64] utilised publicly accessible datasets to evaluate their proposed schemes.

Gu and Gu [59] introduced the inverse term entropy (ITE), which applies to multi-class and binary classification. Moreover, it is computationally inexpensive compared to STW schemes devised based on statistics. The authors evaluated the proposed ITE scheme using two multi-class and two binary-class datasets. In a study [65], an enhanced version of TF-IGM, called TF-IGM_{imp}, was introduced, along with an enhanced version of RTF-IGM, referred to as RTF-IGM_{imp}. The results of the experiment showed that the proposed TF-IGM_{imp} and RTF-IGM_{imp} outperformed standard TF-IGM and RTF-IGM, respectively. TF-DFS, a novel STW scheme, was proposed by Chen et al. [66] and is based on a well-known approach for feature selection called distinguishing feature selector (DFS). The authors also proposed TF-MDFS, a modified version of TF-DFS, to address the flaws observed in TF-DFS. Overall, the results of their experiment demonstrated that the TF-MDFS outperformed the advanced weighting schemes.

2.8 Research Gap

According to Table 2.3 and the preceding explanation, all works on term weighting in ML-based EQC are based on USTW schemes. Although STW schemes have shown success in text classification, there is a lack of previous research evaluating their effectiveness in BT-based examination questions. However, there is a likelihood that STW schemes may reduce misclassification. Hence, this study evaluated the efficacy of STW methods for classifying BT-based examination questions.

The aforementioned discussion highlights the fact that ETF-IDF and TFPOS-IDF allocated the highest weight to verbs, followed by nouns and adjectives. However, questions may contain multiple types of verbs, such as action verbs and supporting verbs. An example of this can be seen in the below sample question:

"Argue the case for conducting experimental research involving humans and propose guidelines to ensure that the dignity and welfare of the subjects are maintained."

Here, the word 'Argue' is an action verb while the verbs 'conducting,' 'involving,' 'ensure,' and 'maintained' are supporting verbs. The past schemes of EQC, such as ETF-IDF and TFPOS-IDF, do not differentiate between the different types of verbs. This differentiation, however, may result in increased classification accuracy as action verbs play a significant role in determining the cognitive levels of questions compared to supporting verbs. Therefore, the aim of this study is to identify the action verbs in questions and assign them a higher weight compared to the supporting verbs for enhanced classification.

CHAPTER 3

METHODOLOGY

3.1 Research Phases

This research is structured into three phases, as illustrated in Figure 3.1, each designed to align with specific objectives. Phase 1 is dedicated to the comprehensive exploration of Objective 1, Phase 2 covers Objective 2, and Phase 3 focuses on attaining Objective 3. This division facilitates a systematic and in-depth examination of each objective, enabling a comprehensive understanding and contributions in the field of EQC based on BT.

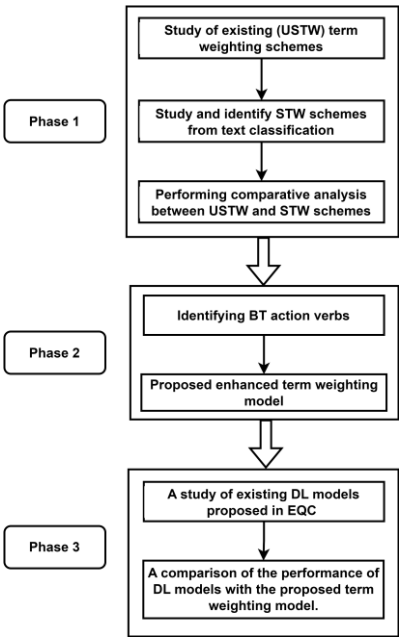

Figure 3.1: Phases involved in this study

3.1.1 Phase 1

3.1.1.1 Study of Existing (USTW) Term Weighting Schemes

In past work, researchers proposed many term weighting schemes in the context of EQC based on BT. These term weighting schemes come under USTW schemes and were reviewed in phase 1 of this study as preliminary work. The objective of reviewing existing term weighting schemes was to gain a better understanding of term weighting in the context of BT-based question classification. As a result, it helped to enhance the term weighting scheme.

3.1.1.2 Study and Identify Supervised Schemes from Text Classification

In this phase, the baseline STW schemes were studied from text classification. The aim was to identify the STW schemes which can be compared with existing USTW schemes used in EQC. Given that there are schemes designed for both binary and multiclass classification, it was imperative to undertake a rigorous study and analysis to determine the most suitable schemes.

3.1.1.3 Performing Comparative Analysis between Unsupervised and Supervised Schemes

The preliminary study of the term weighting schemes in the context of the EQC based on BT was performed. This analysis was carried out to facilitate a comparison between STW and USTW schemes. The term weighting schemes identified in this phase were utilised in this comparison. The outcome of this phase determined whether the STW approach proves effective or ineffective in the context of EQC based on BT.

3.1.2 Phase 2

3.1.2.1 Identifying Bloom's Taxonomy Action Verbs

The BT action verbs are present in the questions that need to be identified in this phase. These BT verbs appear in various positions within the questions. Hence, the questions were manually studied to identify the possible positions of BT verbs. Finally, an algorithm was developed to identify BT action verbs from questions automatically.

3.1.2.2 Proposed Enhanced Term Weighting Model

After conducting a preliminary analysis of the term weighting schemes, this study proposed an enhanced term weighting model to reduce the misclassification of examination questions based on BT. In this phase, ML classifiers were trained for EQC using the enhanced term weighting model. Following that, the trained classification models were evaluated to investigate the performance of the proposed term weighting model.

3.1.3 Phase 3

3.1.3.1 A Study of Existing Deep Learning Models Proposed in Examination Question Classification

In this research phase, the primary aim was to comprehensively examine and analyse the architectures of previously proposed DL models within the field of EQC. This investigation was crucial for gaining insights into the design and structure of these models, facilitating their implementation with the datasets utilised in this study.

3.1.3.2 A Comparison of the Performance of Deep Learning Models with the Proposed Term Weighting Model

In this research phase, an investigation was conducted to compare the performance of DL models with the proposed term weighting model. A two-fold evaluation approach was utilised to assess the effectiveness of DL models. Small-weight DL models underwent rigorous evaluation through both cross-validation and train-test split approach. In contrast, heavy-weight DL models were exclusively evaluated using the train-test split methodology. The goal was to investigate whether the proposed domain-specific term weighting model can outperform DL models in the context of limited dataset size.

3.2 Methods

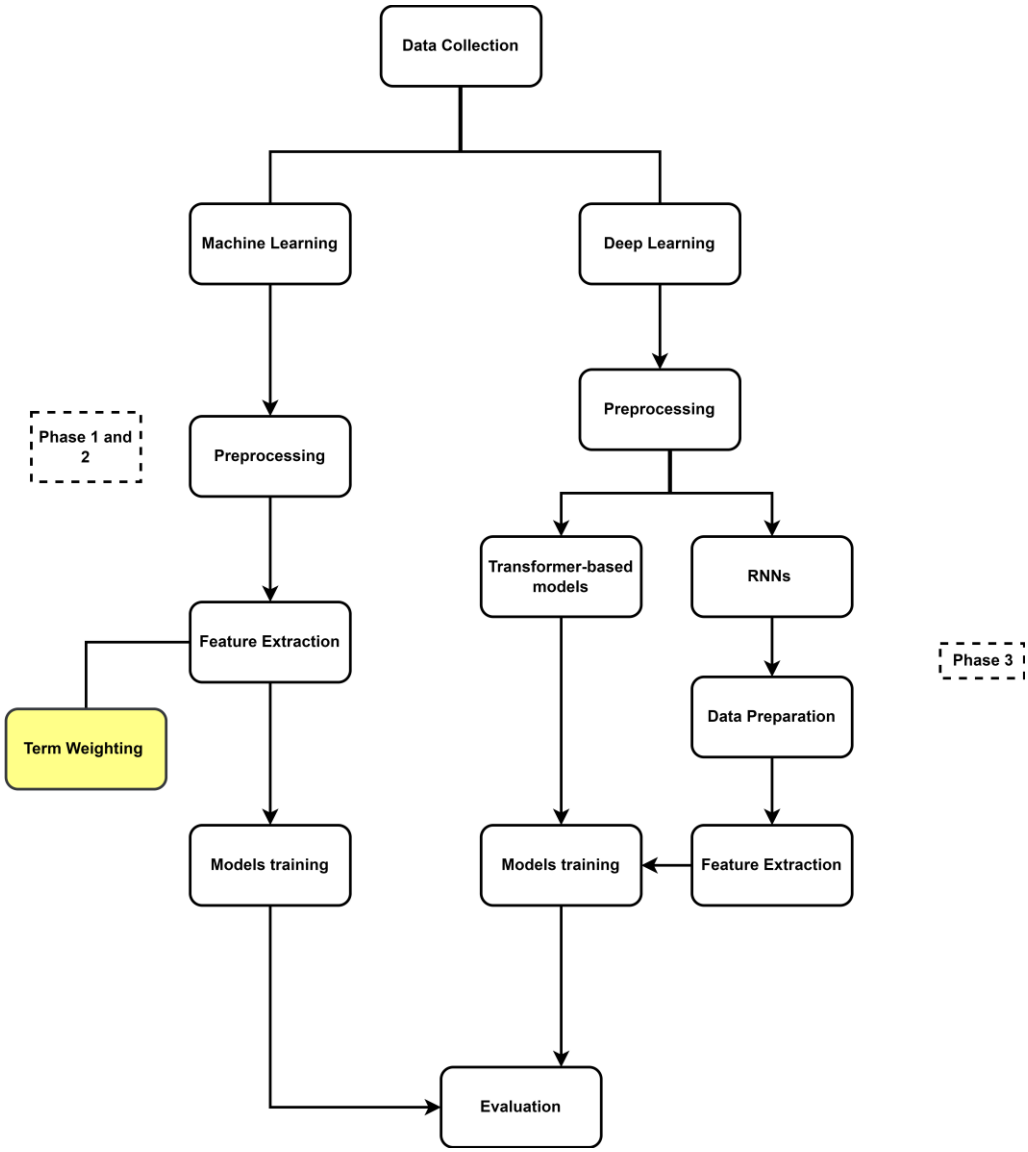

Figure 3.2: Proposed examination question classification model

This chapter outlines the processes required to achieve the research objectives specified in the introduction chapter. The proposed research methodology is presented in Figure 3.2, commencing with the data collection phase. The proposed methodology is divided into two parts: ML and DL. The first two objectives of this study have covered the ML in EQC, whereas the third objective covered the DL in EQC. The first objective utilised STW from text

classification and existing USTW of EQC, whereas the second one used the proposed enhanced term weighting scheme. In addition, the BT keywords were identified from the questions for the second objective, which was not performed for the first objective. The methodology involved in the first two objectives is shown together since both are ML-based and except for dissimilarities, as mentioned earlier, the rest of the steps are identical. The DL-based differs from ML-based since the preprocessing involved in DL-based is different. In addition, the data preparation needs to be carried out only for DL-based classification. However, the evaluation process used for both ML-based and DL-based classification is the same, as shown in Figure 3.2.

3.2.1 Data Collection and Annotation

In this research, a total of five datasets [1] were used, four of which are distinct datasets, and the fifth is a combined form of these four distinct datasets. The combined dataset is used to evaluate the performance of DL models and compare them with the proposed term weighting model. In addition, the combined dataset was used to compare the performance of the proposed model with the existing term weighting model. Among the four distinct datasets, three were collected from past studies of EQC. The first two datasets in Table 3.1 are taken from Sangodiah et al. [12] and the third from Yahya et al. [10]. The third dataset initially contains 600 questions. However, all the questions that do not contain at least one of the BT action verbs were dropped. The fourth dataset was collected from the Universiti Tunku Abdul Rahman (UTAR) library, which comprised 711 questions.

[1] https://doi.org/10.6084/m9.figshare.22597957.v3

Table 3.1: Class distribution of all datasets

Cognitive Level	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Combined Dataset
Knowledge	23	50	56	149	278
Comprehension	37	135	92	180	444
Application	29	72	62	100	263
Analysis	30	56	45	99	229
Synthesis	29	45	66	76	216
Evaluation	33	57	66	107	263
Total	181	415	387	711	1693

The datasets utilised from past studies were already annotated by pedagogy experts according to the BT cognitive domain. Dataset 4, collected from the UTAR library, was annotated by UTAR and Quest International University pedagogy experts. All the questions were labelled to any of the six cognitive levels of BT's cognitive domain. The Dataset 4 was initially consisted of 1200 questions. After annotation, it was found that the dataset is imbalanced as the "Comprehension" level contains way higher questions than other levels. Hence, questions were randomly dropped from the comprehension level to make the dataset as balanced as possible. In Table 3.1, if we look at the class distribution of the datasets, we can see that the class distributions are pretty balanced. Table 3.2 shows some labelled questions from each cognitive level of BT's cognitive domain. Notably, Dataset 1 contains questions from the business domain only; however, Dataset 2 contains questions from multiple domains: Programming, Science, Social Science, Computing, Multimedia, Mathematics, and Business. Furthermore, the authors of Dataset 3 obtained questions from an item bank accessible over the Internet. As per Dataset 4, the questions were collected from multiple domains: Business, Engineering, Science, and Social.

Table 3.2: Sample questions of each cognitive level from the Dataset 3

Cognitive Level	Sample Question
Knowledge	"List three characteristics that are unique to the Cubist movement."
	"Identify fractions from a pictorial representation."
Comprehension	"Explain the whole method of crushing."
	"State in your own words the rule for balls and strikes in baseball."
Application	"Relate the principle of reinforcement to classroom interactions."
	"Construct the 4-Bit Ripple carry Adder Circuit."
Analysis	"Compare fall and spring."
	"Analyse the characteristics of frogs."
Synthesis	"Design costumes for the characters."
	"Compose a simple rap or rhyme about zoo animals."
Evaluation	"Evaluate appropriate and inappropriate actions of characters."
	"Appraise the speech's effectiveness based on the class criteria."

3.2.2 Machine Learning Approach

3.2.2.1 Preprocessing

Figure 3.3 illustrates all the steps involved in the processing of examination questions. Prior to undergoing any preprocessing, all questions from each dataset were transformed into lowercase. Upon completion of the lowercase conversion, the preprocessing procedures consisted of tokenisation and elimination of punctuation marks, POS tagging, BT action verbs identification, removal of stop words, and lemmatisation. However, the BT action verbs identification was carried out only for the proposed term weighting model and skipped for comparison between the STW and USTW schemes. This was due to the fact that the POS of the words alone was sufficient for weighting

during the comparison between the STW and USTW schemes. The output of each step is illustrated in Table 3.3 with the help of the following question:

"Explain why a less experienced multimedia developer (including non-programmers) always opts for multimedia authoring tools to easily create an interactive application."

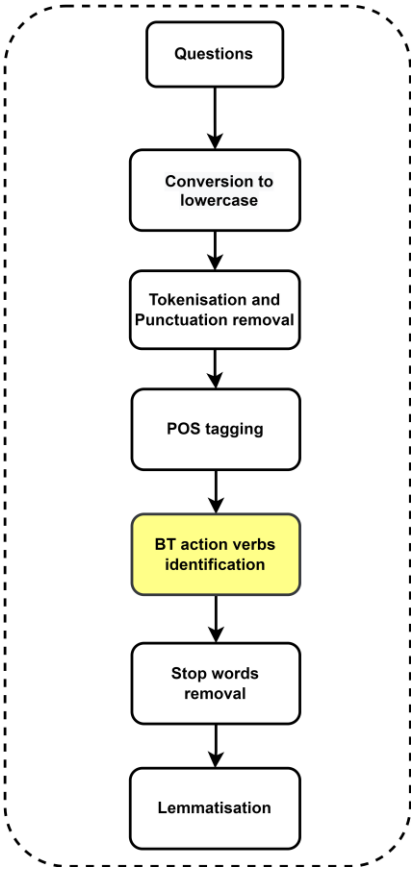

Figure 3.3: Processing steps involved in training machine learning models

Table 3.3: Output of each preprocessing step

Process	Output	Remarks
"Lowercase Conversion"	"explain why a less experienced multimedia developer (including non-programmers) always opts for multimedia authoring tools to easily create an interactive application."	The question was converted to lowercase.
"Tokenisation and Punctuation removal"	['explain' 'why' 'less' 'experienced' 'multimedia' 'developer' 'including' 'non-programmers' 'always' 'opts' 'for' 'multimedia' 'authoring' 'tools' 'to' 'easily' 'create' 'an' 'interactive' 'application']	The question tokenised and removed punctuation: () and period.
"POS Tagging"	[('explain' 'VB') ('why' 'WRB') ('less' 'RBR') ('experienced' 'JJ') ('multimedia' 'JJ') ('developer' 'NN') ('including' 'VBG') ('non-programmers' 'NNS') ('always' 'RB') ('opts' 'VBZ') ('for' 'IN') ('multimedia' 'JJ') ('authoring' 'NN') ('tools' 'NNS') ('to' 'TO') ('easily' 'RB') ('create' 'VB') ('an' 'DT') ('interactive' 'JJ') ('application' 'NN')]	NN = Noun (singular) DT = Determiner TO = Infinitive 'to.' VB = Verb JJ = Adjective VBG = Verb (gerund/ present participle) VBZ = Verb (3rd person singular/present) IN = Preposition and NNS = Noun (plural) RB = Adverb RBR = Adverb (comparative) WRB = WH-adverb
"BT action verbs identification"	[('explain' 'BT') ('why' 'WRB') ('less' 'RBR') ('experienced' 'JJ') ('multimedia' 'JJ') ('developer' 'NN') ('including' 'VBG') ('non-programmers' 'NNS') ('always' 'RB') ('opts' 'VBZ') ('for' 'IN') ('multimedia' 'JJ') ('authoring' 'NN') ('tools' 'NNS') ('to' 'TO') ('easily' 'RB') ('create' 'VB') ('an' 'DT') ('interactive' 'JJ') ('application' 'NN')]	The BT keyword 'explain' at the beginning of the question was identified.
"Stop words removal."	[('explain' 'BT') ('why' 'WRB') ('less' 'RBR') ('experienced' 'JJ') ('multimedia' 'JJ') ('developer' 'NN') ('including' 'VBG') ('non-programmers' 'NNS') ('always' 'RB') ('opts' 'VBZ') ('multimedia' 'JJ') ('authoring' 'NN') ('tools' 'NNS') ('easily' 'RB') ('create' 'VB') ('interactive' 'JJ') ('application' 'NN')]	Stop words: 'for,' 'to,' and 'an' were removed.
"lemmatisation"	[('explain' 'BT') ('why' 'WRB') ('less' 'RBR') ('experience' 'JJ') ('multimedia' 'JJ') ('developer' 'NN') ('include' 'VBG') ('non-programmers' 'NNS') ('always' 'RB') ('opt' 'VBZ') ('multimedia' 'JJ') ('author' 'NN') ('tool' 'NNS') ('easily' 'RB') ('create' 'VB') ('interactive' 'JJ') ('application' 'NN')]	Lemmatised the following words: 'experienced,' 'including,' 'opts,' 'authoring,' and 'tools.'

Tokenisation and Punctuation Removing:

In this study, multiple tokenising techniques were evaluated, including the Python split function, the NLTK library (3.6.1) [67], the TextBlob library (0.17.1) [68], and regular expression. Results indicated that the Python split function, NLTK, and TextBlob returned punctuation as tokens, as demonstrated in Table 3.4. Consequently, additional cleaning steps were necessary to remove the punctuation when using these techniques to tokenise examination questions. As a result, the regular expression method was adopted in this study to tokenise the questions, as it was found to be the most optimal technique among the four considered. The regular expression was constructed to only return words from the questions, eliminating the need for further processing to remove punctuation. The outcome of this step yields a list of words, as depicted in Table 3.4.

In cases where the question consisted of multiple sentences, the regular expression was applied to each sentence individually. Upon applying the regular expression to a sentence, a period (.) was added to the list before proceeding to the next sentence. This approach allows for the distinction between sentences by utilising the period, which is required later to identify the BT action verbs in the question.

Table 3.4: Techniques tested for tokenising questions

Question	Tokenizer	Tokenised Question
"Explain how editor-in-chief can use 6-porter's techniques in the 'manufacturing' industries."	Pyhton Split Function	['explain' 'how' 'editor-in-chief' 'can' 'use' "6-porter's" 'techniques' 'in' 'the' " 'manufacturing' " 'industries.']
	NLTK	['explain' 'how' 'editor-in-chief' 'can' 'use' '6-porter' " 's" 'techniques' 'in' 'the' "'manufacturing" " ' " 'industries' '.']
	TextBlob	['explain' 'how' 'editor-in-chief' 'can' 'use' '6-porter' "'s" 'techniques' 'in' 'the' "'manufacturing" 'industries']
	Regular Expression	['explain' 'how' 'editor-in-chief' 'can' 'use' 'porter' 'techniques' 'in' 'the' 'manufacturing' 'industries']

Part of Speech Tagging:

In this study, it was deemed essential to determine the POS for each word in the examination questions, as the proposed term weighting model required the assignment of weights based on the POS. The NLTK library, TextBlob library, and Stanford Tagger (4.2.0) [69] were tested in this study. The result shows that NLTK and TextBlob tagger tagged the BT action verbs incorrectly in many cases, as illustrated in Table 3.5. For example, in the first question of Table 3.5, the keyword 'analyse' is a BT action verb and a verb; however, the NLTK tagged

it as an adverb, and TextBlob tagged it as a noun. The Stanford tagger tagged correctly only among the three taggers. In the second question, the keyword 'briefly' is an adverb, but the NLTK and TextBlob tagged it as a noun, whereas the Stanford tagger correctly tagged it as an adverb. Due to its superior performance, the Stanford tagger was employed in this study to tag the examination questions despite its higher time complexity than the other two taggers.

Bloom's Taxonomy Action Verbs Identification:

All the datasets used in this study were thoroughly studied and analysed to identify possible patterns in the questions and identify the BT action verbs from the questions. From the questions, some patterns were found to identify the BT action verbs, as demonstrated in Table 3.6. There was no other means to identify the BT action verbs except by manually studying and analysing the positions of the verbs in the questions since no past study of EQC addressed this issue. Therefore, the BT action verbs were identified using their five positions, as illustrated in Table 3.6. This procedure was highly laborious and time-intensive.

Table 3.5: Techniques tested for parts of speech tagging

Question	Tagger	Tagged Question
"Analyse each emerging trend, its relevance, and importance to the designing of future enterprise systems."	NLTK	[('analyse' 'RB') ('each' 'DT') ('emerging' 'VBG') ('trend' 'NN') ('its' 'PRP$') ('relevance' 'NN') ('and' 'CC') ('importance' 'NN') ('to' 'TO') ('the' 'DT') ('designing' 'NN') ('of' 'IN') ('future' 'JJ') ('enterprise' 'NN') ('systems' 'NNS')]
	TextBlob	[('analyse' 'NN') ('each' 'DT') ('emerging' 'VBG') ('trend' 'NN') ('its' 'PRP$') ('relevance' 'NN') ('and' 'CC') ('importance' 'NN') ('to' 'TO') ('the' 'DT') ('designing' 'VBG') ('of' 'IN') ('future' 'NN') ('enterprise' 'NN') ('systems' 'NNS')]
	Stanford	[('analyse' 'VB') ('each' 'DT') ('emerging' 'VBG') ('trend' 'NN') ('its' 'PRP$') ('relevance' 'NN') ('and' 'CC') ('importance' 'NN') ('to' 'IN') ('the' 'DT') ('designing' 'NN') ('of' 'IN') ('future' 'JJ') ('enterprise' 'NN') ('systems' 'NNS')]
"Briefly discuss any two efforts that organisation may perform in order to discourage unethical behavior."	NLTK	[('briefly' 'NN') ('discuss' 'VBZ') ('any' 'DT') ('two' 'CD') ('efforts' 'NNS') ('that' 'IN') ('organisation' 'NN') ('may' 'MD') ('perform' 'VB') ('in' 'IN') ('order' 'NN') ('to' 'TO') ('discourage' 'VB') ('unethical' 'JJ') ('behavior' 'NN')]
	TextBlob	[('briefly' 'NN') ('discuss' 'VB') ('any' 'DT') ('two' 'CD') ('efforts' 'NNS') ('that' 'IN') ('organisation' 'NN') ('may' 'MD') ('perform' 'VB') ('in' 'IN') ('order' 'NN') ('to' 'TO') ('discourage' 'VB') ('unethical' 'JJ') ('behavior' 'NN')]
	Stanford	[('briefly' 'RB') ('discuss' 'VB') ('any' 'DT') ('two' 'CD') ('efforts' 'NNS') ('that' 'DT') ('organisation' 'NN') ('may' 'MD') ('perform' 'VB') ('in' 'IN') ('order' 'NN') ('to' 'TO') ('discourage' 'VB') ('unethical' 'JJ') ('behavior' 'NN')]

Table 3.6: Positions of Bloom's Taxonomy action verbs in the questions

Position of BT action verb	Question	BT action verb	Supporting verb
"The first word of a question."	"Write a short story relating a personal experience in the style of a picaresque novel."	Write	Relating
"The second word of a question, followed by an adjective."	"Briefly discuss why the emergence of the data warehouse phenomenon is causing such interest in the business world."	Discuss	Causing
"The second word of a question, followed by an adjective, and after the conjunction AND."	"Critically appraise the five competitive forces encased within Porter's "FIVE FORCES" model within the context of a profit-oriented organisation and discuss the threat posed to the firm by each of these forces."	Appraise, Discuss	Encased, Posed
"The first word of a question, and after the conjunction AND."	"Identify one problem in the book and give an alternate solution one not given by the author."	Identify, Give	Given
"Joined by the conjunction AND."	"Compare and contrast animals that the class has made."	Compare, Contrast	Made

Figure 3.4 shows the algorithm which was implemented to identify the BT action verbs from the questions. The algorithm was designed to identify action verbs from all five positions listed in Table 3.6. Each question was passed into the function, and the function identified the BT action verbs and returned the results. The input of the function is the POS-tagged questions and BT action verbs database. The BT action verbs database was collected from a past study [12].

In this process, the initial step involves extracting the first word from the question passed into the function. The first word is then verified against the BT action verbs database to determine if it is a BT action verb. If it is identified as a BT action verb, it is appended to a new list with the label ‘BT’; otherwise, it is appended with the same label it received during the POS tagging process. The remaining words are checked against the database to determine if they are BT action verbs. If they are not, they are appended to the list with their original label. However, if the word is present in the database, the preceding word is evaluated to determine if it is the conjunction “and” or a period (.). If either of these conditions is met, the word is appended to the list with the label ‘BT.’ If not, the preceding word is further examined to determine if it is an adverb ending with “ly.” If this condition is met, the word is appended to the list with the label ‘BT’; otherwise, it is appended with the original label obtained through POS tagging.

Stop Words Removal:

After identifying the BT action verbs, the stop words present in the questions were eliminated using the NLTK stop word list. However, some of the stop words were removed from the NLTK stop word list before eliminating stop words from the questions. These stop words include ‘how,’ ‘why,’ ‘between,’ ‘what,’ ‘which,’ and ‘who.’ The reason for not eliminating these words is that all of these stop words were widely used in the questions and might impact the cognitive level.

```
q: A Question
d: BT Action Keywords Database
function IDENTIFY(q, d)
    newlist ← []
    x ← first word of q
    rw ← words of q except x
    if x is in d then
        newlist.insert((x, "BT"))
    else
        newlist.insert((x, x.POS))
    end if
    for word in rw do
        if word is in d then
            y ← previous word
            if y == ("and" or Full stop(.)) then
                newlist.insert((word, "BT"))
            else if y is (Adverb and Ends with "ly") then
                newlist.insert((word, "BT"))
            else
                newlist.insert((word, y.POS))
            end if
        else
            newlist.insert((word, y.POS))
        end if
    end for
    return newlist
end function
```

Figure 3.4: Algorithm to identify the Bloom's Taxonomy action verbs

Lemmatisation:

This study evaluated the NLTK and spaCy (3.4.1) [70] lemmatisers to determine the optimal lemmatising technique. Results showed that while the NLTK lemmatiser was generally accurate, it produced incorrect results in instances where the noun did not end with "ing." In contrast, the spaCy lemmatiser performed correctly for such nouns. A combination of both lemmatisers was implemented through a custom Python function to address this issue. Additionally, it was found that specifying the POS tag for the words being lemmatised was crucial to prevent erroneous results, as the NLTK lemmatiser defaulted to considering all words as nouns.

3.2.2.2 Feature Extraction

The feature extraction process encompasses creating a feature set and the implementation of the term weighting model, as depicted in Figure 3.5. In this study, the unigram technique was utilised for extracting features from the preprocessed data. Regarding term weighting, this study investigated the STW for EQC and proposed a novel term weighting model, which was evaluated against the existing term weighting scheme of EQC. The outcome of this process will serve as the ML classifier's input during the EQC model's training.

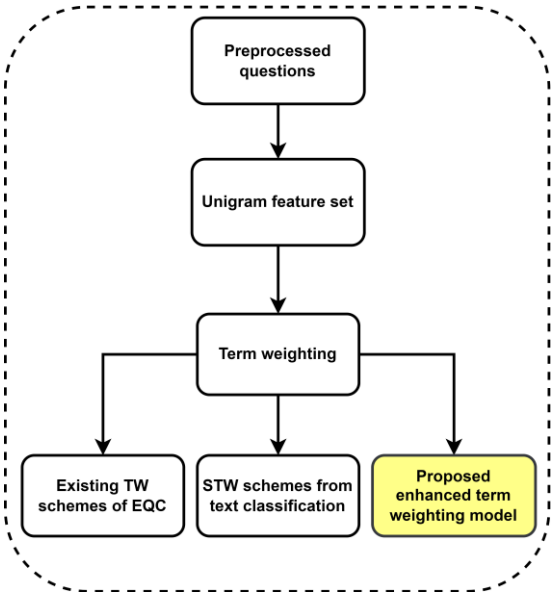

Figure 3.5: Feature extraction steps involved in training machine learning models

Feature Set:

The unigram technique is a basic method for feature extraction, which involves creating a set of all unique words present in the dataset. Previous studies [7], [13], [71] have utilised the unigram approach to extract the feature set from examination questions. Aside from unigram, several other feature extraction

techniques exist, such as bigram, trigram, POS tagging, headword, and others, as reported by Sangodiah et al. [71]. The primary feature extraction method was used due to the fact that the focus of the study was to improve term weighting for the EQC task. Hence, the unigram feature set was extracted from each dataset used in this study.

Low-frequency terms can be crucial in classifying examination questions; hence, feature selection could result in the loss of significant features. Therefore, no feature selection was performed in this study following the extraction of the feature set. This is in line with the approach taken by Sangodiah et al. [12], which also refrained from feature selection due to concerns of losing valuable features that infrequently appear in questions. Numerous past studies of EQC [7], [13], [71] on term weighting also did not perform feature selection. An additional reason for not conducting feature selection is the small size of the dataset, as the creation of a large dataset of examination questions presents significant challenges and demands time.

Term Weighting:

This section discussed the term weighting schemes implemented in this study. These term weighting schemes include three existing USTW schemes of EQC, three base STW schemes from text classification, and the proposed enhanced term weighting model ETFPOS-IDF. The existing USTW schemes of EQC are compared with the STW schemes from text classification in terms of performance as well as with the proposed term weighting model. The existing USTW schemes implemented in this study are TF-IDF, ETF-IDF [7], and

TFPOS-IDF [13]. Additionally, the base STW schemes implemented are TF-ICF [61], TF-IDF-ICF [62], and TF-IDF-ICSDF [62].

Existing Term Weighting Schemes of Examination Question Classification:

TF-IDF: According to Sangodiah et al. [55], numerous variations of TF-IDF are used in text classification and EQC. The variant used in the study [13] differs from the one used in [33]. After a thorough analysis, Sangodiah et al. [55] determined that the variant utilised in the study [13] was the most optimal among the different variations of TF-IDF. As a result, this study adopted this variant of TF-IDF. The formulae for TF and IDF are presented in Equations (1) and (2), respectively.

$$TF\ (t, q) = \frac{C(t_q)}{T_q} \tag{1}$$

In (1), $C(t_q)$ is the number of times term t appears in question q, and T_q represents the total number of terms in question q.

$$IDF(t) = 1 + log\ (\frac{Q}{q_t}) \tag{2}$$

In (2), Q is the total number of questions in the dataset, and q_t is the total number of questions that contain the term t. Finally, *TF-IDF (t, q)* is the multiplication of the (1) and (2), as shown below in (3).

$$TF\text{-}IDF\ (t, q) = TF\ (t, q).IDF(t) \tag{3}$$

ETF-IDF: The traditional TF-IDF was improved for EQC in the study [7] with the introduction of the impact factor (IF), as demonstrated in (4), which provides the formula for its calculation. The IF was allocated to terms based on their POS categorisation.

$$IF(t) = \begin{Bmatrix} X(t) + 3, \ if \ t \ is \ VB \\ X(t) + 1, \ if \ t \ is \ NN \ or \ ADJ \\ 1, \ otherwise \end{Bmatrix} \quad (4)$$

In (4), X*(t)* is,

$$X(t) = \sqrt{\frac{1}{C}\sum_{i=1}^{C}(eq\ (t, c_i) - \frac{1}{C})^2} \quad (5)$$

In (5), *C* represents the total number of classes in the dataset, which is six in this study, since the BT cognitive domain consists of six distinct levels. The equation $eq\ (t, c_i)$ refers to the total number of questions that belong to the class c_i and contains the term *t*, divided by the overall number of questions in the dataset. Finally, the *ETF-IDF (t, q)* was obtained by multiplying *TF-IDF (t, q)* by the *IF (t)*, as represented in (6).

$$E\text{-}TFIDF\ (t, q) = TF\text{-}IDF\ (t, q).\ IF(t) \quad (6)$$

TFPOS-IDF: The authors of the study [7] introduced TFPOS-IDF in their later study [13]. The TFPOS-IDF is an enhanced version of the standard TF-IDF that incorporates POS-based weighting.

$$w_{pos}(t) = \begin{Bmatrix} w1 \ \ if \ t \ is \ verb \\ w2 \ \ if \ t \ is \ noun \ or \ adjective \\ w3 \ \ otherwise \end{Bmatrix} \quad (7)$$

In (7), the value for *w1* = 5, *w2* = 3, and *w3* = 1. The formula to calculate the TFPOS *(t, q)* is shown in (8).

$$TFPOS\ (t, q) = \frac{C(t,q) * w_{pos}(t)}{\sum_i C(t_i,q) * w_{pos}(t_i)} \quad (8)$$

In (8), $C(t,\ q)$ refers to the total number of times term *t* presents in question *q*, and $C(t_i, q)$ represents the frequency of each term in question *q*. Finally, the *TFPOS-IDF (t, q)* was calculated with (9) by multiplying the *TFPOS (t, q)* and

the *IDF(t)*.

$$TFPOS\text{-}IDF\ (t, q) = TFPOS\ (t, q).IDF(t) \quad (9)$$

As shown in (10), the TFPOS-IDF was combined with a word embedding approach referred to as Word2vec. The purpose of utilising Word2vec was to obtain semantic information, and the authors claimed that it effectively addressed the issue of BT overlapping keywords.

$$Question\ vector = \sum_{t \epsilon q} Word2vec(t) * [TFPOS - IDF\ (t, q)] \quad (10)$$

Supervised Term Weighting from the Text Classification:

TF-ICF: This STW scheme was proposed in the study [61]. The formula to calculate the TF-ICF is given in (11). The findings of the study [55] reported the optimal variant of TF, which is identical to the one used in TF-ICF.

$$TF\text{-}ICF\ (t_i, q_j) = tf\ (t_i, q_j) * log\ (1 + \frac{|C|}{cf(t_i)}) \quad (11)$$

In (10), the raw TF of the term t_i in question q_j is represented as $tf\ (t_i, q_j)$. The number of classes in the dataset is indicated by the symbol $|C|$. The number of classes in which the term t_i presents is denoted by $cf(t_i)$.

TF-IDF-ICF: This STW scheme was proposed by Ren and Sohrab [62]. In this scheme, the TF-ICF is combined with the IDF by taking products of them. However, the ICF used in this scheme is not exactly the same as the TF-ICF proposed in the study [61]. The equation to calculate the TF-IDF-ICF is given in (12).

$$TF\text{-}IDF\text{-}ICF\ (t_i, q_j) = TF\text{-}IDF\ (t_i, q_j) * (1 + log\ \frac{C}{c(t_i)}) \quad (12)$$

In (12), the TF used in *TF-IDF* (t_i, q_j) is the raw TF, and IDF is the same one discussed earlier in (2). The total number of classes in the dataset is denoted as *C*. The number of classes in which the term t_i occurs is represented by $c(t_i)$.

TF-IDF-ICSDF: This term weighting scheme was also introduced by Ren and Sohrab [62]. In this scheme, the ICF is replaced with the ICSDF by the authors, and the TF-IDF portion of this scheme remains identical to the TF-IDF-ICF.

$$TF\text{-}IDF\text{-}ICSDF\ (t_i, q_j) = TF\text{-}IDF\ (t_i, q_j) * \left(1 + log \frac{C}{\sum_{k=1}^{C} \frac{d(t_i, c_k)}{d(c_k)}}\right) \quad (13)$$

In (13), the number of questions that belong to class c_k is represented by $d(c_k)$. The number of questions belongs to class c_k and contains the term t_i is symbolised by $d(t_i, c_k)$.

Proposed Term Weighting Model ETFPOS-IDF:

This study enhanced the TFPOS-IDF by introducing the different weighting for different types of verbs and named ETFPOS-IDF. The existing schemes, such as TFPOS-IDF and ETF-IDF, introduced POS-based weighting and assigned different weights for different POS, and the highest weight was assigned to the verbs. This way of weighting the terms shows improvement in EQC based on BT. However, the TFPOS-IDF and ETF-IDF considered all the verbs in the questions equally significant. Questions may contain more than one type of verb: BT action verbs and supporting verbs, as discussed earlier. The BT action verbs have a higher impact on determining the BT levels while categorising the questions according to BT than the supporting verbs. Therefore, this study differentiated between BT action verbs and supporting verbs, with the

former receiving a higher weight due to their higher impact on determining BT levels. This way of discriminating between verbs helps improvement in the EQC. Since the ETFOS-IDF needs BT action verbs to be present in the questions, this scheme can only be applied in BT-based classification, such as EQC and designing learning outcomes. This makes the proposed term weighting model specific for the BT-based classification. The formulae to calculate ETFPOS-IDF are provided in (14) to (16).

$$Ew_{pos}(t) = \begin{Bmatrix} w1 \ \ if\ t\ is\ BT\ action\ verb \\ w2\ if\ t\ is\ supporting\ verb \\ w3 \ \ if\ t\ is\ noun\ or\ adjective \\ w4 \ \ otherwise \end{Bmatrix} \tag{14}$$

In (14), The weight of the BT action verb (w1) = 5, the weight of the supporting verb (w2) = 3, the weight of the noun and adjective (w4) = 2, and 1 for the rest of the POS. The weight difference between the BT action verb and the supporting verb used in this study is 2; however, this is not the optimal weight difference. This study aimed to investigate whether the different weights for BT action verbs and supporting verbs increase classification accuracy. Therefore, the optimal weight was not calculated here since it depends on the data. The optimal weight may vary from dataset to dataset, making it a hyperparameter that needs to be tuned for each dataset.

$$ETFPOS\ (t,\ q) = \frac{C(t,q) * Ew_{pos}(t)}{\sum_i C(t_i,q) * Ew_{pos}(t_i)} \tag{15}$$

In (2), the equation to calculate the *ETFPOS (t, q)* is illustrated. In the dividend, the TF of *t* in question *q* is represented by $C(t,\ \ q)$. In the denominator, the frequency of each of the terms present in question q is multiplied by their $Ew_{pos}(t)$ and summing the resulting products. Finally, the *ETFPOS-IDF (t, q)* is the product of *ETFPOS (t, q)* and *IDF(t)*, as presented in (16).

$$ETFPOS\text{-}IDF\ (t, q) = ETFPOS\ (t, q).IDF(t) \quad (16)$$

According to [13], the normalisation technique is helpful during model training to avoid numerical complexity in calculations. Therefore, in this study, the *L2* normalisation was applied to the proposed ETFPOS-IDF scheme. The *L2* normalisation converted all the weighting values between 0 and 1. In (17), the formula, which was used to normalise the weighting values, is given.

$$Normalized\ ETFPOS\text{-}IDF(t, q) = \frac{ETFPOS-IDF(t,q)}{\sqrt{\sum ETFPOS-IDF(t,q)^2}} \quad (17)$$

In (17), the dividend *ETFPOS-IDF(t, q)* represents the weighting value of term *t* in question *q*. The denominator involves taking the sum of the squared term weighting values for all terms in question *q*, followed by computing the square root of this sum.

3.2.2.3 Model Training

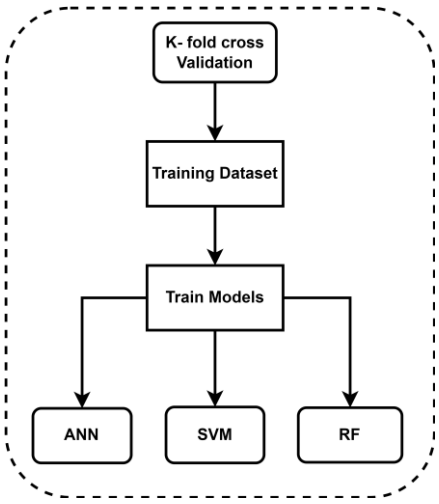

Figure 3.6: Steps involved in training machine learning models

Figure 3.6 shows the training steps of the ML models trained using the STW and USTW schemes and the proposed term weighting model using three ML classifiers.

Support Vector Machine:

SVM is a supervised ML algorithm initially proposed by Vladimir Vapnik and his colleagues [72]. SVM has found extensive applications in both text classification and EQC [9], [33], which is known for its superior text classification accuracy [12]. The goal of SVM is to learn an optimal hyperplane, which is a linear decision boundary that effectively segregates the two sets of data, as shown in Figure 3.7. In the case of EQC, the two sets of data represent the different types of questions. SVM attempts to identify a hyperplane that can segregate the two sets of questions with maximum efficiency. To achieve this, SVM supports multiple kernels such as linear, polynomial, sigmoid, and radial basis functions. Previous works on EQC [10], [13], [71] have frequently utilised the linear kernel of SVM. Therefore, the current study employed the linear kernel of SVM to train the model. The advantage of using SVM for EQC is its ability to generalise well in high-dimensional feature spaces. Similar to text classification, EQC also deals with high-dimensional feature spaces, and SVM is known for its ability to handle such spaces effectively [52].

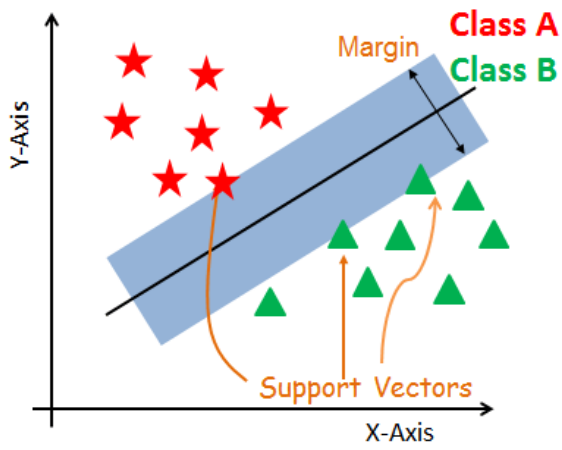

Figure 3.7: The illustration of the support vector machine classifier [73]

Random Forest:

The RF classifier, introduced by Leo Breiman [74], has been identified as one of the most effective classifiers for text classification [75]. This classifier is widely utilised [63], [76], [77] to solve many text classification problems. RF is based on decision trees and employs an ensemble learning technique where each decision tree predicts a class. Therefore, it utilises majority voting to decide the final predicted class [75], as illustrated in Figure 3.8. A notable advantage of RF is its capability to address the overfitting problem [78], which has been a concern in the decision tree classifier. In this study, the RF implemented by the Scikit-learn library was utilised with the default settings and a random state of 42 to ensure the reproducibility of the results.

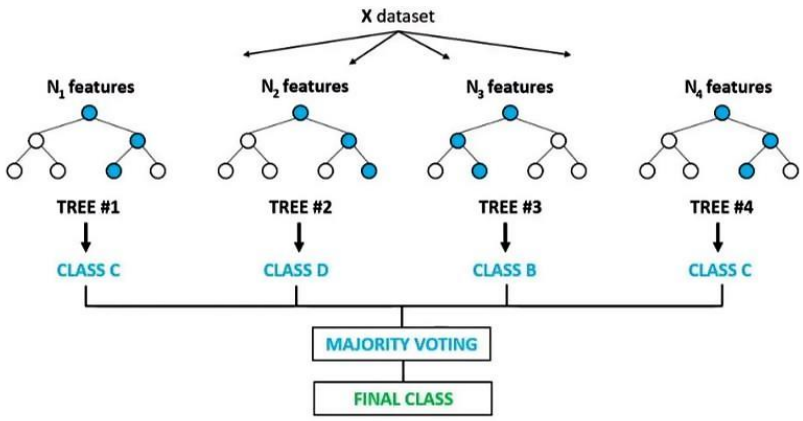

Figure 3.8: The illustration of the random forest classifier [79]

Artificial Neural Network:

The Multilayer Perceptron classifier is an ML algorithm that is inspired by the structure and function of the human brain. This algorithm, also referred to as ANN, has been utilised in numerous previous studies [80], [81] to classify text data. The ANN comprises the input, hidden, and output layers, as illustrated in Figure 3.9. Each layer contains multiple nodes connected to nodes in the previous layers. In ANN, Information is transmitted between the nodes through connections, which are assigned weights that determine the strength of the connection. An ANN can include more than one hidden layer. However, for this study, the default settings for the number of hidden layers and neurons in the ANN classifier available in Scikit-learn were utilised. The default hidden layers and neurons in the Scikit-learn implementation are 1 and 100, respectively. As a random state for ANN, zero was used to achieve reproducible results. Besides this, ‘lbfgs’ was used as a solver since it converges faster with the small dataset, according to Scikit-learn [82] documentation. The default activation function used in the Scikit-learn implementation of ANN is ReLU, which was used in this

study. In addition to the activation function, the default setting was used for the optimiser, which is ‘Adam.’

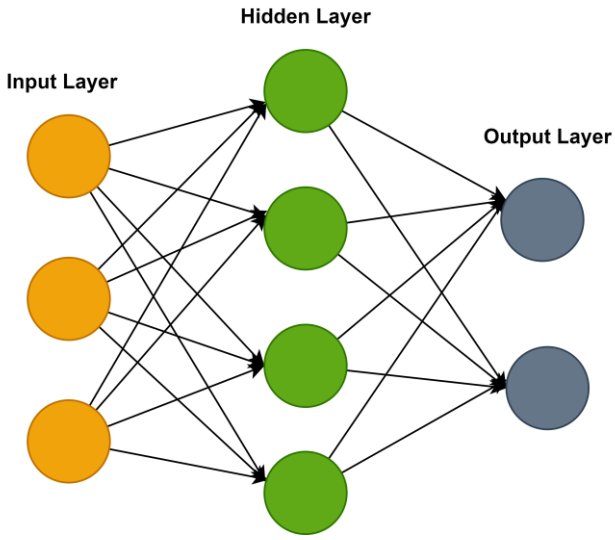

Figure 3.9: The illustration of an artificial neural network with one hidden layer

3.2.3 Deep Learning Approach

In this research, two recurrent neural networks (RNNs), which are LSTM and gated recurrent unit (GRU), and transformer-based models proposed in past studies of EQC were compared with the term weighting model proposed in this study. The GRU has never been tested for EQC-based BT. Hence, this study used the GRU with the exact experiment settings of LSTM used by an earlier study [5]. This study followed all the steps performed by the earlier study for LSTM: preprocessing, data preparation, feature extraction, and training. Regarding the transformer-based models, this study compared the fine-tuned BERT proposed by Das et al. [48] and BERT + DNN proposed by Sharma et al. [4] and followed the experiment settings utilised in those studies.

3.2.3.1 Recurrent Neural Networks

Preprocessing and Data Preparation:

Figures 3.10 and 3.11 show the data preprocessing and data preparation, respectively. The examination questions were appropriately formatted by following a few preprocessing steps. Preprocessing examination questions involves conversion to lowercase, tokenisation, punctuation and stop word removal, and lemmatisation. Data preparation involves creating vocabulary from the training set and converting each question into a sequence based on the index assigned to each of the unique words present in the vocabulary. Following this, padding was applied by appending zeroes to each sequence and extending them to 500 words [5]. The padding ensured that all the sequences maintained the same length.

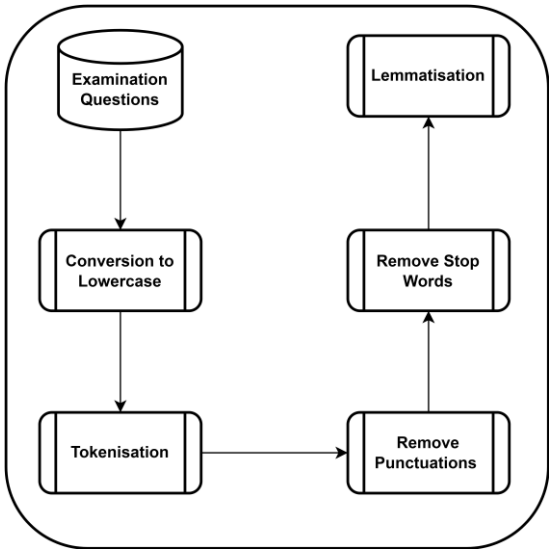

Figure 3.10: Preprocessing steps involved in training recurrent neural networks

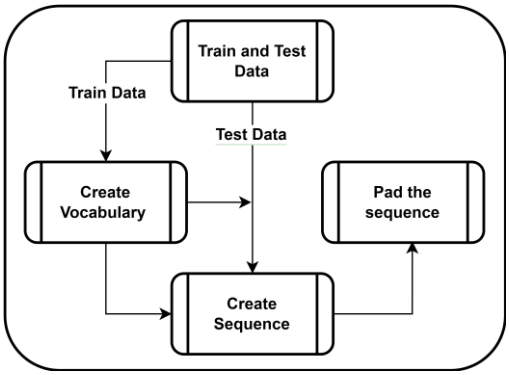

Figure 3.11: Data preparation steps involved in training recurrent neural networks

Features:

In the present research, the embedding layer was not trained from scratch by following the past study [5]. Instead, embeddings were derived from the pre-trained model FastText for each term in the vocabulary constructed during the preprocessing phase, as shown in Figure 3.12. These embeddings were subsequently transferred to the embedding layer as initial weights during training.

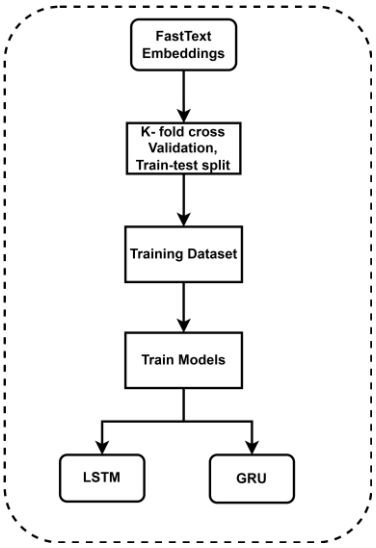

Figure 3.12: Steps involved in training recurrent neural networks

Training:

Figure 3.13 shows LSTM layers with 32 neurons. The dense output layer consisted of 6 neurons and used softmax activation. Before the output layer, a dropout layer with a value of 0.2 was used to reduce overfitting. In addition, the architecture of the GRU model is also identical except for the GRU layer instead of LSTM. These models were trained with a batch size of 16 and utilised the RMSprop optimiser with the default learning rate.

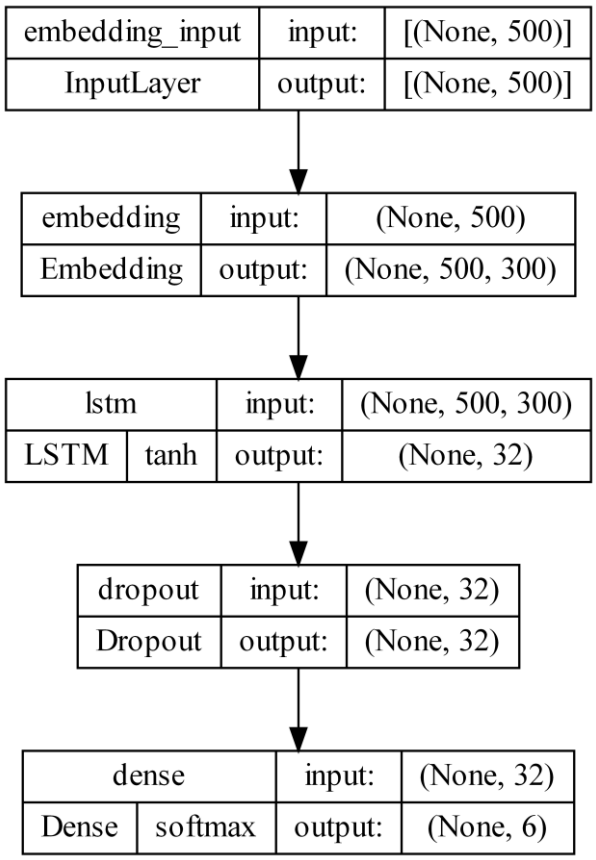

Figure 3.13: The architecture of the long short-term memory model

3.2.3.2 Transformer-Based Models

Fine-tuning is a process where pre-trained models are further trained on specific tasks to adapt them to specific domains or applications. In this study, the fine-tuning model proposed in earlier studies of EQC is compared with the proposed term weighting model.

Preprocessing:

The examination questions were preprocessed using the preprocessor available with the pre-trained models. The processing steps performed by the preprocessor involve vocabulary mapping, masking, adding special tokens 'CLS' and 'SEP', padding the sequences, and more. The preprocessor inserts the special token 'CLS' at the beginning of the input sequence and the token 'SEP' between sentences to distinguish multiple sentences within the same input sequence. The 'SEP' token is also placed at the end of the input sequence.

Fine-Tuning:

Figure 3.14 shows the steps involved in fine-tuning BERT. The pre-trained model BERT was initialised with the pre-trained weights before fine-tuning. The pre-trained BERT was fine-tuned without any changes in the architecture by following the past study [48] and another time by adding a DNN, as suggested by Sharma et al. [4]. Subsequently, the models undergo additional training using a labelled EQC dataset. RMSprop with a learning rate of 3e-5 and a batch size of 16 were employed to optimise the training, and the models were fine-tuned for 5 epochs.

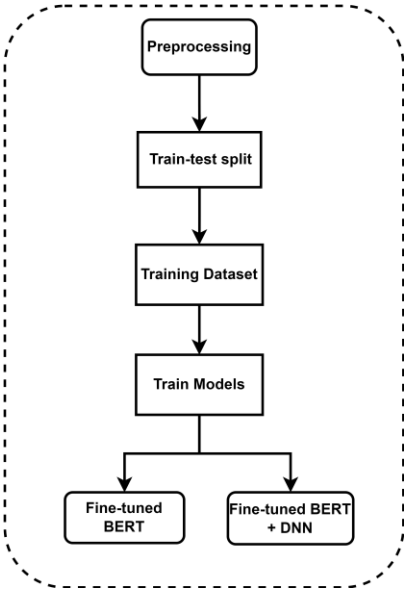

Figure 3.14: Steps involved in fine-tuning BERT

3.2.4 Evaluation

Figure 3.15 illustrates all the steps in evaluating the models trained in this study.

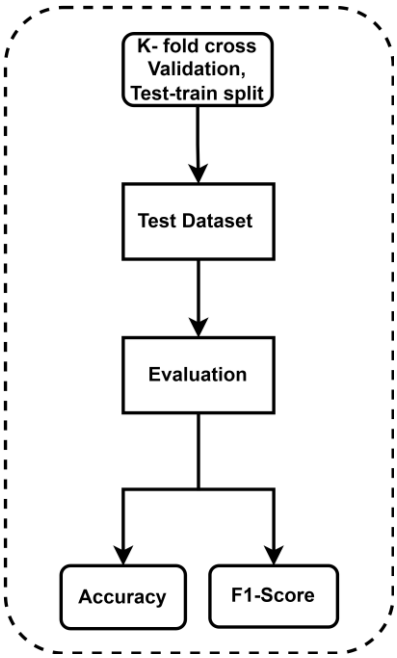

Figure 3.15: Steps involved in evaluating the machine and deep learning models

3.2.4.1 K-Fold Cross-Validation

Cross-validation is a resampling technique for evaluating ML models on a limited sample of data. This study used the k-fold cross-validation technique to split the dataset into train and test for model training and testing. In many past works of EQC [7], [8], k-fold cross-validation was used for the abovementioned purpose. It splits the dataset into k folds where k is a positive integer. The k-fold cross-validation trains the model k-1 times and tests or evaluates using the remaining fold. This process is iterative, repeats k times, and records the performance metrics of each iteration. The final result is calculated by taking the average of each iteration. Figure 3.16 illustrates the k-fold cross-validation technique with k = 5. From Figure 3.16, at the first iteration, 'Fold 1' is used for testing and the rest of the folds for training. In the second iteration, 'Fold 2' is utilised for testing and the rest for training, and a similar strategy is followed for the rest of the iterations. The advantage of using cross-validation over the train-test split is that it gives a better estimate at the cost of more computation [83] and reveals inconsistencies that can signal overfitting. Consequently, this study analyses the acquired training and test metrics values to detect possible overfitting.

The Stratified k-fold cross-validation, a k-fold cross-validation variation, was utilised in this study. The reason for using stratified k-fold cross-validation is that it makes sure that each fold has the same proportion of observations belonging to a specific class [84]. In this study, multiple k-values were used following a past study [71] of EQC for more reliable performance measurement. The k-values used were 3 to 10. At first, the average performance of each k-

value was calculated, as shown in (18), since each k-value consists of a number of k-iterations (refer to Figure 3.16). After that, the average performance of all k-values was computed, as shown in (19).

$$\overline{A_k} = \frac{\sum_{i=1}^{k} A_i}{k} \tag{18}$$

In (18), the symbol $\overline{A_k}$ represents the average score obtained by each k-fold value, where the symbol A_i is the score obtained by a particular iteration. The score of each iteration was added up and divided by k to get the average.

$$\overline{A} = \frac{\sum_{k=3}^{10} \overline{A}_k}{\sum_{3}^{10} 1} \tag{19}$$

In (19), $\overline{A}$ indicates the average score of all k-values, where a total of 8 k-values was used in this study. The average scores of each k-value were added together, and the final average was calculated by dividing the sum by 8.

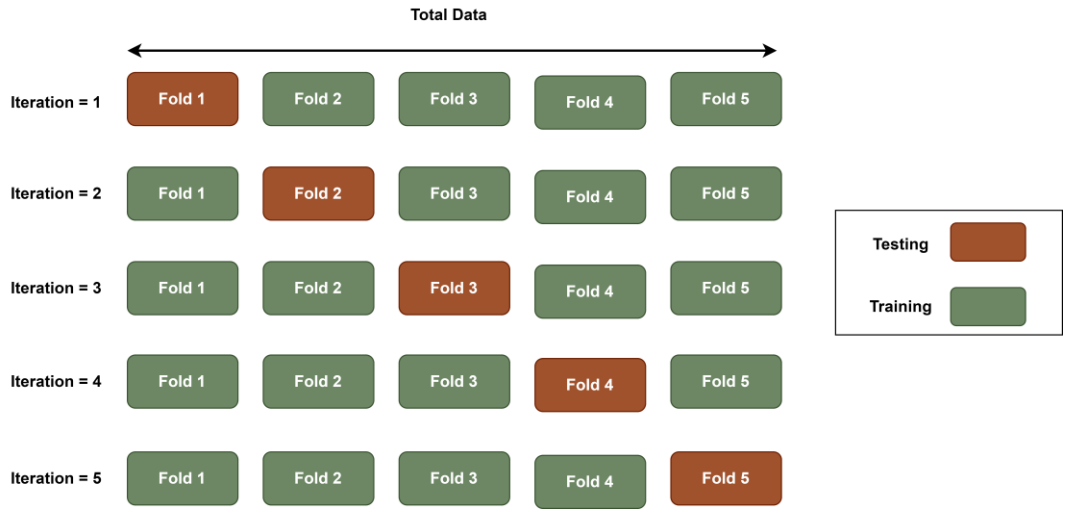

Figure 3.16: An illustration of k-fold cross-validation with k = 5

In this study, the Sciki-learn implementation of stratified cross-validation was used. Using a fixed random state in cross-validation ensures that the same result is produced consistently across multiple runs. As stated in the Scikit-learn documentation [82], a random state can be specified as an integer value. Therefore, a random state of 0 was chosen and utilised in this scenario.

3.2.4.2 Train-Test Split

Along with cross-validation, the train-test approach was also utilised in this study. The train-test split technique was used in two scenarios only, comparing the RNNs with the proposed term weighting model and the transformer-based model with the proposed term weighting model. Though RNNs were evaluated with cross-validation along with the train-test split, transformer-based models were only evaluated with the train-test split. Transformer-based models require significant computational resources and time to train. Cross-validation involves training and evaluating the model multiple times, which can be prohibitively expensive in terms of time and resources. A train-test split allows you to assess the model's performance with just one round of training, making it more practical for large models like BERT. The dataset was split into the training and testing sets, where 90% of the data was in the training set and 10% in the test set. The small dataset size is the reason for using 10% for testing rather than the commonly used 20%. In addition, the stratified splitting process was followed to ensure an equal proportion of questions from each cognitive level in both the train and test sets.

3.2.4.3 Evaluation Metrics

This study utilised the accuracy [85] and macro F1 score [86] as evaluation metrics. Numerous past studies [10], [33], [37] of EQC utilised these metrics to evaluate performance. To calculate the macro averaged F1 score, first need to calculate the F1 score of each class and then calculate the mean of all classes. The reason for using the macro version of the F1 score is that not every dataset used in this study is fully balanced. The macro average considers each class equally important and better estimates the performance when the datasets are not balanced. The formula for accuracy is given in (20), the F1 score in (21), and the macro F1 score in (22).

$$Accuracy = \frac{Total\ correct\ predictions}{Total\ predictions} \tag{20}$$

Accuracy is the proportion of correct predictions made by the classifier. It is calculated as the ratio of correct predictions to the total number of predictions, as shown in (20).

$$F1\ score = 2\ x\ \frac{Precision\ x\ Recall}{Precesion + Recall} \tag{21}$$

In (21), *Precision = TP / (TP + FP)*, and *Recall = TP / (TP+ FN)*. *TP* represents the true positive, and *FP* and *FN* represent the false positive and false negative, respectively. The total number of correctly predicted positive instances is *TP*. *FP* corresponds to the number of instances predicted as positive but actually not positive, while *FN* is the number of instances predicted as negative but actually positive.

$$Macro\ F1\ score = \frac{\sum_1^C F1\ score}{C} \tag{22}$$

In Equation (22), *C* refers to the number of classes in the test set. The F1 score of each of the classes is added up and divided by the total number of the classes.

CHAPTER 4

RESULTS AND DISCUSSION

This chapter discusses the results and findings of this study in a comprehensive manner. The chapter is subdivided into three sections: a comparison of USTW and STW, the performance of the proposed term weighting model, and a comprehensive comparison of the proposed term weighting model with DL models.

4.1 Supervised Versus Unsupervised Schemes

This section analysed the results of STW and USTW schemes evaluated in this study. These USTW schemes are standard TF-IDF, ETF-IDF, and TFPOS-IDF, whereas STW schemes are TF-ICF, TF-IDF-ICF, and TF-IDF-ICSDF. A comprehensive evaluation was conducted to assess the performance of these term weighting schemes in the context of EQC based on BT, utilising four distinct datasets and three classifiers: SVM, RF, and ANN.

4.1.1 Results of Support Vector Machine

Table 4.1 illustrates the performance of the STW and USTW schemes utilising the SVM classifier. Within the USTW schemes, TFPOS-IDF demonstrated superior performance across all four datasets compared to ETF-IDF and TF-IDF. Among the STW schemes, TF-ICF yielded the highest accuracy and F1 score. A comparative analysis of the STW and USTW schemes

reveals that TF-ICF outperformed all other schemes in three datasets, while TFPOS-IDF excelled in Dataset 4. Though in Datatset 1, the STW scheme TF-IDF-ICF outperformed the USTW scheme ETF-IDF, overall, the USTW scheme ETF-IDF demonstrated superior performance compared to the TF-IDF-ICF. Notably, the STW scheme TF-IDF-ICSDF exhibited the least satisfactory performance among all the schemes. Regarding the average results, it is evident that the STW scheme, TF-ICF, outperformed all other schemes, with the USTW scheme, TFPOS-IDF, following closely behind.

Table 4.1: Results of support vector machine

	USTW Schemes						STW Schemes					
Term weighting/ Dataset	TF-IDF		ETF-IDF		TFPOS-IDF		TF-ICF		TF-IDF-ICF		TF-IDF-ICSDF	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Dataset 1	0.698	0.684	0.713	0.707	0.733	0.731	**0.777**	**0.775**	0.746	0.739	0.672	0.658
Dataset 2	0.629	0.616	0.684	0.680	0.689	0.680	**0.691**	**0.684**	0.664	0.654	0.584	0.557
Dataset 3	0.733	0.729	0.798	0.795	0.807	0.804	**0.810**	**0.807**	0.784	0.781	0.713	0.708
Dataset 4	0.763	0.768	0.812	0.813	**0.813**	**0.816**	0.809	0.810	0.787	0.791	0.699	0.704
AVG	0.706	0.699	0.752	0.749	0.761	0.758	**0.772**	**0.769**	0.745	0.741	0.667	0.657

4.2.2 Results of Random Forest

The performance of USTW and STW schemes with RF classifiers are tabulated in Table 4.2. The findings indicate that within all four datasets, the USTW scheme TFPOS-IDF consistently outperforms the other two USTW schemes, namely ETF-IDF and standard TF-IDF. Among the STW schemes, TF-ICF demonstrated superior performance in three datasets. In contrast, in Dataset 4, TF-IDF-ICF surpassed the other two schemes, and TF-IDF-ICSDF

outperformed TF-ICF. When comparing STW and USTW schemes, it is evident that, across three datasets, the USTW scheme TFPOS-IDF outperforms all other schemes. However, it is worth noting that in Dataset 1, the STW scheme TF-ICF scheme outperformed TFPOS-IDF by a slight margin, with an accuracy difference of 0.3% and an F1 score difference of 0.5%. According to the average result, the USTW scheme TFPOS-IDF outperformed all, followed by the STW scheme TF-ICF. In line with the results obtained for the SVM classifier, TF-IDF-ICSDF demonstrated the least satisfactory performance among all the schemes.

Table 4.2: Results of random forest

	USTW Schemes						STW Schemes					
Term weighting/ Dataset	TF-IDF		ETF-IDF		TFPOS-IDF		TF-ICF		TF-IDF-ICF		TF-IDF-ICSDF	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Dataset 1	0.694	0.692	0.688	0.684	0.702	0.699	**0.705**	**0.704**	0.696	0.693	0.701	0.699
Dataset 2	0.661	0.649	0.661	0.649	**0.689**	**0.680**	0.659	0.647	0.655	0.643	0.647	0.633
Dataset 3	0.741	0.736	0.740	0.735	**0.768**	**0.764**	0.751	0.747	0.748	0.745	0.736	0.732
Dataset 4	0.814	0.816	0.820	0.821	**0.826**	**0.827**	0.812	0.813	0.815	0.816	0.814	0.815
AVG	0.728	0.723	0.727	0.722	**0.746**	**0.743**	0.732	0.728	0.729	0.724	0.724	0.720

4.2.3 Results of Artificial Neural Network

Table 4.3 presents the results obtained using ANN for different term weighting approaches: USTW and STW. Within USTW schemes, it is noteworthy that ETF-IDF consistently demonstrated superior performance compared to both TFPOS-IDF and standard TF-IDF across all four datasets utilised in this study. Among the three classifiers used in this study, ETF-IDF outperformed TFPOS-IDF only with ANN. Between the STW schemes, TF-ICF

outperformed all other schemes in all datasets. This outcome is consistent with the outcome of the SVM classifier. If we compare the STW and USTW schemes, it is found that overall, the STW scheme TF-ICF outperformed all other schemes, followed by the USTW scheme ETF-IDF. In line with the findings observed with the SVM and RF classifiers, the ANN analysis also indicated that the STW scheme TF-IDF-ICSDF yielded the least satisfactory results among all the schemes evaluated in this study.

Table 4.3: Results of artificial neural network

	USTW Schemes						STW Schemes					
Term weighting/ Dataset	TF-IDF		ETF-IDF		TFPOS-IDF		TF-ICF		TF-IDF-ICF		TF-IDF-ICSDF	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Dataset 1	0.697	0.686	0.714	0.702	0.707	0.695	**0.735**	**0.727**	0.694	0.681	0.634	0.626
Dataset 2	0.670	0.667	0.701	0.698	0.696	0.693	**0.722**	**0.718**	0.698	0.692	0.640	0.630
Dataset 3	0.751	0.747	0.806	**0.804**	0.780	0.778	**0.807**	0.802	0.778	0.775	0.714	0.709
Dataset 4	0.758	0.758	**0.809**	**0.809**	0.794	0.794	0.795	0.795	0.767	0.768	0.716	0.714
AVG	0.719	0.715	0.756	0.753	0.744	0.740	**0.765**	**0.761**	0.734	0.729	0.676	0.670

4.1.4 Summary

Figure 4.1 summarises the results of the USTW and STW schemes evaluated in this study. The results depicted in Figure 4.1 correspond to the mean performance of each respective scheme. These mean values were computed by averaging the results across all classifiers and datasets utilised in this research. The results showed that the STW scheme TF-ICF outperformed all other schemes followed by the USTW scheme TFPOS-IDF. The difference in performance between TF-ICF and TFPOS-IDF is very nominal, approximately

0.6% in accuracy and 0.7% in F1 score. Furthermore, the results showed that the USTW schemes TFPOS-IDF and ETF-IDF outperformed STW schemes TF-IDF-ICF and TF-IDF-ICSDF. Additionally, in both evaluation metrics, accuracy, and F1 score, the standard TF-IDF scheme outperformed the STW scheme TF-IDF-ICSDF.

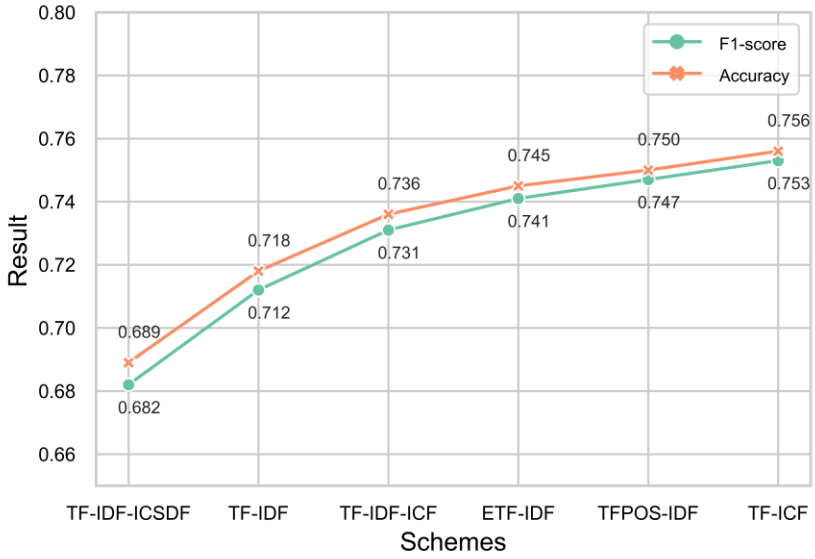

Figure 4.1: Summary of the results

4.2 Performance of the Proposed Term Weighting Model

This section evaluated the performance of the proposed term weighting model ETFPOS-IDF with five different datasets and three classifiers: SVM, ANN, and RF. Additionally, the performance of the proposed term weighting model is compared with the existing schemes of the EQC, which are TF-IDF, ETF-IDF, and TFPOS-IDF.

4.2.1 Results of Support Vector Machine

Table 4.4: Results of support vector machine

Term weighting/ Dataset	TF-IDF		ETF-IDF		TFPOS-IDF		ETFPOS-IDF (Proposed)	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Dataset 1	0.698	0.684	0.713	0.707	0.733	0.731	**0.750**	**0.748**
Dataset 2	0.629	0.616	0.684	0.680	0.689	0.680	**0.700**	**0.696**
Dataset 3	0.733	0.729	0.798	0.795	0.807	0.804	**0.843**	**0.840**
Dataset 4	0.763	0.768	0.812	0.813	0.813	0.816	**0.836**	**0.837**
AVG	0.706	0.699	0.752	0.749	0.761	0.758	**0.782**	**0.780**

Table 4.4 shows the performance of the proposed term weighting model and existing schemes with SVM in all four datasets used in this study. The experiment results showed that the proposed ETFPOS-IDF outperformed TF-IDF, ETF-IDF, and TFPOS-IDF across all datasets. In the Dataset 2 dataset, the difference in performance between the proposed ETFPOS-IDF and TFPOS-IDF is approximately 1% in accuracy and 1.5% in F1 score. However, in other datasets, the difference is higher. If we look at the average performance of each scheme with SVM as the classifier, we can notice that the proposed ETFPOS-IDF achieved 0.782 in accuracy and 0.780 in F1 score. The proposed ETFPOS-IDF outperformed the standard TF-IDF with approximately 8% in both accuracy and F1 score, the ETF-IDF with around 3%, and 2% with TFPOS-IDF. These outcomes indicate that the proposed ETFPOS- IDF improves the classification accuracy of EQC with SVM.

4.2.2 Results of Random Forest

Table 4.5: Results of random forest

Term weighting/ Dataset	TF-IDF		ETF-IDF		TFPOS-IDF		ETFPOS-IDF (Proposed)	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Dataset 1	0.694	0.692	0.688	0.684	0.702	0.699	**0.707**	**0.704**
Dataset 2	0.661	0.649	0.661	0.649	**0.689**	**0.680**	0.672	0.662
Dataset 3	0.741	0.736	0.740	0.735	0.768	0.764	**0.791**	**0.788**
Dataset 4	0.814	0.816	0.820	0.821	0.826	0.827	**0.830**	**0.831**
AVG	0.728	0.723	0.727	0.722	0.746	0.743	**0.750**	**0.746**

The experiment results of the proposed term weighting model ETFPOS-IDF and existing schemes with RF classifier are tabulated in Table 4.5. The results show that the proposed ETFPOS-IDF outperformed the existing TF-IDF and ETF-IDF schemes in all four datasets used in this study. The proposed ETFPOS-IDF also outperformed the TFPOS-IDF in three datasets. In Dataset 2, the TFPOS-IDF outperformed the proposed ETFPOS-IDF by a slight margin. However, the proposed ETFPOS-IDF performed slightly higher than TFPOS-IDF in the other three datasets. Overall, the proposed ETFPOS-IDF achieved an average of 0.750 and 0.746 in accuracy and F1 score, respectively, with the RF classifier, which is approximately 2% higher than the performance of TF-IDF and ETF-IDF. However, with the TFPOS-IDF, the difference is less than 1% in both accuracy and F1 score. In summary, the outcomes of RF are aligned with the outcomes of SVM except with Dataset 2.

4.2.3 Results of Artificial Neural Network

Table 4.6: Results of artificial neural network

Term weighting/ Dataset	TF-IDF		ETF-IDF		TFPOS-IDF		ETFPOS-IDF (Proposed)	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Dataset 1	0.697	0.686	0.714	0.702	0.707	0.695	**0.736**	**0.732**
Dataset 2	0.670	0.667	0.701	0.698	0.696	0.693	**0.715**	**0.714**
Dataset 3	0.751	0.747	0.806	0.804	0.780	0.778	**0.833**	**0.832**
Dataset 4	0.758	0.758	0.809	0.809	0.794	0.794	**0.811**	**0.811**
AVG	0.719	0.715	0.756	0.753	0.744	0.740	**0.774**	**0.772**

Table 4.6 presents the experiment results of the proposed term weighting model ETFPOS-IDF and other existing schemes with the ANN classifier using four different datasets. The results of the experiment show that the proposed ETFPOS-IDF outperformed all three existing schemes: TF-IDF, ETF-IDF, and TFPOS-IDF in all datasets. Among the existing schemes, ETF-IDF performed closely to the ETF-IDF, with approximately 2% lower than the proposed ETFPOS-IDF. The proposed ETFPOS-IDF achieved an average accuracy of 0.774 and an average F1 score of 0.772, which is around 5% higher than the performance of standard TF-IDF in both metrics and 3% higher than the performance of TFPOS-IDF. Overall, the results are consistent with the results of SVM, where the proposed ETFPOS-IDF showed superiority over existing schemes in terms of performance.

4.2.4 Results of Existing Schemes Versus Proposed Model in the Combined Dataset

Figure 4.2 depicts the performance of the ANN classifier in both the train and test sets of the combined dataset. Analysis of the figure reveals that the training accuracy and F1 score exhibit an upward trend with an increasing number of folds in the k-fold cross-validation procedure. This pattern is mirrored in the test set data. Notably, the difference between the metric values observed in the train and test sets remains consistent across various k-fold values. This observation is particularly noteworthy in light of the combined dataset comprising less than 1700 questions and the absence of hyperparameter tuning during model training. Despite these limitations, the model demonstrates a low susceptibility to overfitting and excellent generalization capabilities on unseen data, achieving accuracy and F1 score values exceeding 80% on the test set, albeit slightly lower than the respective train set values of close to 90%.

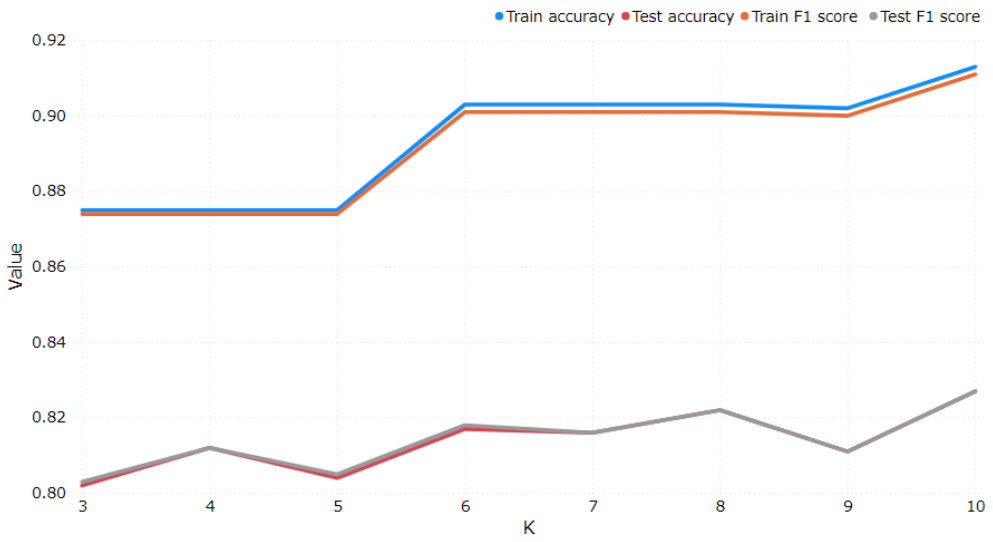

Figure 4. 2: Performance comparison between training and test sets using an artificial neural network

Table 4.7: Results of existing schemes and proposed term weighting model

Term weighting/ Classifier	TF-IDF		ETF-IDF		TFPOS-IDF		ETFPOS-IDF (Proposed)	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SVM	0.788	0.790	0.811	0.813	0.811	0.813	**0.828**	**0.829**
RF	0.808	0.809	0.807	0.809	0.813	0.814	**0.822**	**0.823**
ANN	0.779	0.779	0.812	0.812	0.801	0.801	**0.814**	**0.814**

Table 4.7 presents the performance of the existing and proposed ETFPOS-IDF achieved when applied to the combined dataset. The results demonstrate the superior performance of the proposed ETFPOS-IDF in comparison with the existing scheme across all the classifiers. The proposed ETFPOS-IDF attained higher performance using SVM than with RF and ANN. However, with all the classifiers, the ETFPOS-IDF performed over 80% in both accuracy and F1 score.

4.2.4.1 Statistical Test

This study performed the two-sample t-test to determine whether the difference in performance between ETFPOS-IDF and existing schemes is statistically significant. The results obtained from the combined dataset were used to perform the t-test. The results in Table 4.8 show that the differences in performance between the proposed ETFPOS-IDF and existing schemes are statistically significant, as the p-value is less than the alpha value of 0.05 in all cases. This outcome suggests that the difference in performance is not likely to have occurred randomly or by chance but rather indicates a meaningful and systematic distinction between the proposed ETFPOS-IDF and other existing schemes.

Table 4.8: Statistical test results between existing schemes and proposed term weighting model

Comparison	P-value		Significant?	
	Acc	F1	Acc	F1
TF-IDF vs ETFPOS-IDF	0.000033	0.00006	Yes	Yes
ETF-IDF vs ETFPOS-IDF	0.000077	0.000101	Yes	Yes
TFPOS-IDF vs ETFPOS-IDF	< 0.00001	<0.00001	Yes	Yes

4.2.5 Summary

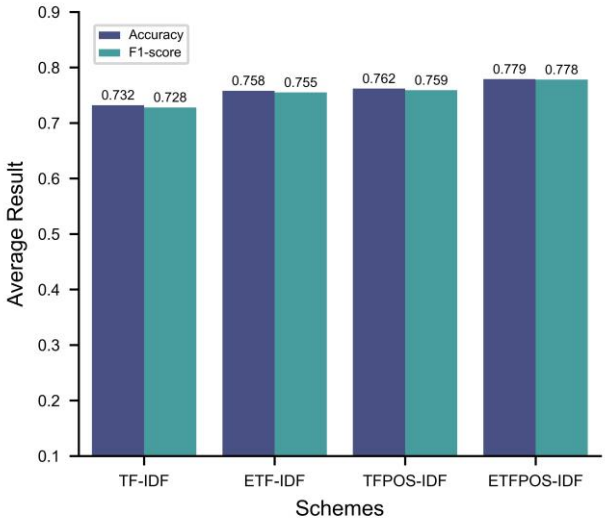

Figure 4.3: Summary of the results

Figure 4.3 illustrates the average performance of each existing scheme along with the proposed ETFPOS-IDF. These results were achieved by calculating the arithmetic mean of the performance of three separate classifiers used in this study. From the figure, it is clear that the proposed ETFPOS-IDF outperformed all three existing schemes and achieved a mean accuracy of 0.779 and an F1 score of 0.778. Among the three existing schemes, the TFPOS-IDF performed near the proposed ETFPOS-IDF. The proposed ETFPOS-IDF

outperformed the TFPOS-IDF and ETF-IDF by a margin of around 2% in both evaluation metrics used in this study. However, the difference is approximately 5% with TF-IDF. Overall, the performance of the proposed ETFPOS-IDF shows improvement when it is used as term weighting for EQC based on BT.

4.3 Proposed Term Weighting Model Versus DL Models

This section compared the performance of the proposed term weighting model with the existing DL models proposed in earlier studies using both cross-validation and train-test split approaches.

4.3.1 Cross-Validation

Table 4.9: Results with cross-validation in the combined dataset

K	LSTM + FastText		GRU + FastText		SVM + Proposed ETFPOS-IDF		RF + Proposed ETFPOS-IDF		ANN + Proposed ETFPOS-IDF	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
3	0.770	0.771	0.753	0.752	0.817	0.819	0.800	0.802	0.802	0.803
4	0.786	0.786	0.765	0.764	0.822	0.824	0.812	0.813	0.812	0.812
5	0.780	0.780	0.764	0.764	0.826	0.828	0.819	0.820	0.804	0.805
6	0.781	0.781	0.767	0.766	0.827	0.829	0.825	0.825	0.817	0.818
7	0.780	0.780	0.763	0.764	0.827	0.828	0.820	0.822	0.816	0.816
8	0.785	0.786	0.760	0.760	0.832	0.833	0.836	0.837	0.822	0.822
9	0.776	0.776	0.765	0.765	0.835	0.837	0.829	0.830	0.811	0.811
10	0.793	0.793	0.767	0.766	0.835	0.836	0.835	0.836	0.827	0.827
AVG	0.781	0.782	0.763	0.763	**0.828**	**0.829**	0.822	0.823	0.814	0.814

Table 4.9 presents the results of various ML and DL models with cross-validation in the combined dataset. The models include LSTM + FastText, GRU + FastText, SVM + proposed ETFPOS-IDF, RF + proposed ETFPOS-IDF, and ANN + proposed ETFPOS-IDF. The evaluation metrics used are accuracy and

F1 score, which are standard metrics to assess the performance of classification models.

Firstly, when comparing the DL models LSTM + FastText and GRU + FastText, it becomes evident that LSTM consistently outperforms GRU in terms of both accuracy and F1 score. This suggests that LSTM might be a superior choice for EQC. Secondly, when comparing the ML models (SVM, RF, and ANN) to the DL models across various 'K' values, it becomes evident that the ML models consistently achieved higher accuracy and F1 scores. This indicates that domain-specific term weighting with ML models is more effective than DL models in this context where the datasets are not extensive.

4.3.1.1 Statistical Test

This study performed the two-sample t-test to determine whether the difference in performance between the proposed term weighting model ETFPOS-IDF and existing DL models is statistically significant. The analysis was based on results derived from the combined dataset using cross-validation. The findings in Table 4.10 demonstrate that the differences in performance between the proposed ETFPOS-IDF and the existing DL models are statistically significant, as evidenced by p-values consistently falling below the predetermined alpha threshold of 0.05 in all instances. This outcome implies that the distinction in performance is unlikely to result from random chance but instead conveys a significant and systematic differentiation between the proposed term weighting model and the other preexisting DL models.

Table 4.10: Statistical test results between existing deep learning models and proposed term weighting model

Comparison	P-value		Significant?	
	Acc	F1	Acc	F1
LSTM vs ETFPOS-IDF	< 0.00001	< 0.00001	Yes	Yes
GRU vs ETFPOS-IDF	< 0.00001	< 0.00001	Yes	Yes

4.3.2 Train-Test Split

Table 4.11: Results with the train-test split in the combined dataset

Work	Technique	Acc	F1
Das et al. (2020)	Fine-tuned BERT	0.806	0.805
Shaikh et al. (2021)	LSTM + FastText	0.753	0.748
-	GRU + FastText	0.741	0.738
Sharma et al. (2022)	Fine-tuned BERT + DNN	0.788	0.787
Proposed model	ANN + ETFPOS-IDF	0.818	0.819
Proposed model	SVM + ETFPOS-IDF	0.847	0.846
Proposed model	RF + ETFPOS-IDF	**0.871**	**0.871**

Table 4.11 compares two approaches: DL methods used in existing research and our proposed ML model using the train-test split technique. In this comparative analysis, this study presents the performance of various techniques employed in the task. This study applied the DL models proposed by earlier studies on the combined dataset. Das et al. (2020) utilised a fine-tuned BERT model, which achieved an accuracy of 0.806 and an F1 score of 0.805 in the combined dataset. The model proposed by Shaikh et al. (2021), which is LSTM with FastText embeddings, attained an accuracy of 0.753 and an F1 score of 0.748. While the exact settings like LSTM were used on GRU, the results

decreased and achieved 0.741 and 0.738 in accuracy and F1 score, respectively. Sharma et al. (2022) incorporated BERT in conjunction with a DNN, which yielded an accuracy of 0.788 and an F1 score of 0.787 in the combined dataset.

In contrast, ML algorithms ANN, SVM, and RF, integrated with ETFPOS-IDF, yielded accuracy and F1 score values of 0.818/0.819, 0.847/0.846, and 0.871/0.871, respectively. It is evident from this analysis that the proposed ETFPOS-IDF with ML algorithms consistently outperformed the prior DL models, demonstrating superior accuracy and F1 scores across the board. This shows that traditional ML methods with domain-specific term weighting can be very effective for EQC tasks, surpassing the more popular DL methods.

4.3.3 Summary

This comprehensive analysis of EQC techniques, encompassing cross-validation and train-test split approaches, reveals several key insights. While DL models, particularly LSTM, excel in specific scenarios, it is notable that traditional ML models, coupled with domain-specific term weighting like ETFPOS-IDF, consistently outperform DL models across both evaluation metrics. This highlights the importance of dataset size and the specific task requirements in choosing the most effective model. It is worth noting that DL models such as fine-tuned BERT and LSTM with FastText embeddings demonstrated competitive performance. Still, they were consistently outperformed by traditional ML models in this task in the context of limited dataset size and when domain-specific term weighting was incorporated as a

feature set in ML algorithms. This challenges the conventional belief that transformer-based models outperform traditional ML in NLP tasks in every scenario. Thus, while BERT and other transformer-based models and RNNs remain potent tools in NLP, our findings emphasise the efficacy of traditional ML methods when applied judiciously with domain-specific feature engineering for specific tasks like EQC.

4.4 Discussion

The first research question of this study is, “Does the STW scheme outperform the USTW scheme for EQC?” Based on the obtained results, it is evident that among the six evaluated schemes, the STW scheme TF-ICF demonstrated superior performance in terms of both accuracy and F1 score when compared to the other five schemes. The TF-ICF scheme comprises TF, which is term frequency, a common component in all schemes examined in this study, and ICF (Inverse Category Frequency). This underscores the influence of category information on the accuracy of classification.

However, it is noteworthy that the USTW schemes, specifically ETF-IDF and TFPOS-IDF, outperformed the other STW schemes, TF-IDF-ICF and TF-IDF-ICSDF. This suggests that the combination of IDF (Inverse Document Frequency) and ICF may not be as effective in EQC. It is essential to highlight that in text classification [62], TF-IDF-ICF exhibited superiority over TF-ICF. However, these evaluations predominantly employed extensive datasets, whereas EQC datasets typically tend to be smaller. In light of this discussion, the

notion that STW schemes consistently outperform USTW schemes is not supported by the findings derived from the EQC dataset.

The second research question is: “Does discriminating between the verb types while term weighting enhance EQC accuracy?” Based on the findings, it has been determined that the proposed ETFPOS-IDF demonstrates superior performance when compared to all existing schemes, namely TF-IDF, ETF-IDF, and TFPOS-IDF. Additionally, statistical tests confirm that the observed performance difference between the proposed ETFPOS-IDF scheme and the existing schemes is statistically significant and not the result of random chance. In ETF-IDF and TFPOS-IDF, a notable emphasis was placed on assigning higher weights to verbs in contrast to other POS, as verbs are considered more pivotal in determining the cognitive levels of examination questions [12], [13]. However, it is important to note that these schemes did not distinguish between different types of verbs, such as supporting and BT action verbs.

Table 4.12: Term weighting values of the proposed ETFPOS-IDF and other schemes (1)

Terms/ Scheme	calculate (BT action verb)	follow (supporting verb)	equation
TF-IDF	0.615629	0.491937	0.615629
ETF-IDF	0.75141	0.599134	0.276443
TFPOS-IDF	0.707367	0.565243	0.42442
ETFPOS-IDF (proposed)	0.848229	0.406681	0.339291

Table 4.13: Term weighting values of the proposed ETFPOS-IDF and other schemes (2)

Terms/ Scheme	list (BT action verb)	step	involve (supporting verb)	titration
TF-IDF	0.39348	0.499809	0.545603	0.545603
ETF-IDF	0.541701	0.253646	0.751998	0.276988
TFPOS-IDF	0.4882	0.372075	0.676942	0.406165
ETFPOS-IDF (proposed)	0.665491	0.338131	0.553666	0.369111

Conversely, the ETFPOS-IDF model introduced a distinction by assigning a greater weight to BT action verbs compared to supporting verbs, a feature not previously addressed in the earlier schemes. Tables 4.12 and 4.13 display the term weighting values for the proposed model and existing schemes using a randomly selected question from Dataset 3. Tables 4.12 and 4.13 illustrate that the distinction in weighting between BT action verbs and supporting verbs is more prominent in the proposed model ETFPOS-IDF when compared to TF-IDF, ETF-IDF, and TFPOS-IDF. As a result, the enhanced performance of ETFPOS-IDF could potentially be attributed to this discrimination between verb types while weighting the terms. Hence, to address the second research question, it can be inferred that discriminating between different types of verbs indeed contributes to an improvement in the accuracy of EQC based on BT.

The third research question is, “Does the proposed term weighting model outperform DL models for EQC based on BT?” The study results demonstrate that the proposed term weighting model performed better than the existing DL models proposed in EQC in the context of limited dataset size. This evaluation

encompassed both train-test splits and cross-validation methodologies. Notably, the statistical analyses revealed that the performance superiority of the proposed term weighting model over the existing DL models is statistically significant. In light of these findings, it is reasonable to conclude that the proposed model consistently outperforms the existing DL models when operating within the constraints of a limited dataset size.

CHAPTER 5

CONCLUSION

5.1 Overview

The main objective of this study was to investigate whether discriminating the different types of verbs present in the examination questions while term weighting has an impact on the classification accuracy or not. In pursuit of this objective, the present study introduced a novel term weighting model denoted as ETFPOS-IDF, incorporating this novel conceptualization. In addition, the study also analyzed whether the STW scheme outperforms the USTW scheme for EQC, as this was not explored in any of the past studies of EQC based on BT. Lastly, the proposed term weighting model was compared with standard DL models proposed in earlier studies to classify examination questions.

The STW schemes from text classification and USTW schemes from EQC were studied and analyzed to achieve the first objective. From the literature, this study identified the widely used three supervised schemes, TF-ICF, TF-IDF-ICF, and TF-IDF-ICSDF, to compare with the existing schemes of EQC: TF-IDF, ETF-IDF, and TFPOS-IDF. To achieve the second objective, the existing scheme TFPOS-IDF was enhanced and modified to assign a different and higher weight to the BT action verbs than the supporting verbs. The BT action verbs must first be identified from the questions. The identification process of BT action verbs was time-consuming and laborious. The datasets used

in this study were studied to identify the possible patterns of the BT action verbs' position. This study identified the patterns, and an algorithm was written based on those patterns to identify the BT action verbs. LSTM, GRU, and BERT were used to achieve the third objective. These are models proposed in past studies to classify examination questions.

This study used four different datasets and three classifiers, SVM, RF, and ANN, to train the models for the first two objectives. Three datasets were taken from the past study, and another was collected from the UTAR library. The UTAR and Quest International University academicians labelled the dataset according to the BT cognitive domain. Four datasets used in the first two objectives were combined into one and used to compare the proposed model's performance with existing schemes and DL models. This study used accuracy and F1 score with stratified K-fold cross-validation to evaluate the models. However, in the third objective, stratified K-fold cross-validation and stratified train-test split were used to evaluate the performance.

The results of the experiment showed that the supervised scheme TF-ICF outperformed all the existing term weighting schemes of EQC. However, the ETF-IDF and TFPOS-IDF outperformed the other supervised term weighting schemes. These outcomes do not indicate the consistent superiority of the supervised term weighting scheme over the unsupervised term weighting scheme and vice versa. The experiment results showed that the proposed ETFPOS-IDF outperformed the existing schemes of EQC in all the classifiers and datasets. This finding demonstrates that distinguishing the verb types for EQC is significant. Comparing the performance of the DL models and the

proposed term weighting model showed that the proposed model outperformed the DL models in the context of limited dataset size.

5.2 Contributions

5.2.1 Theoretical Contributions

- **Impact of verb type discrimination**

This research has primarily contributed theoretically by identifying a previously unexplored research gap: incorporating verb type discrimination into term weighting to classify examination questions. Moreover, this study has addressed whether verb type discrimination influences the accuracy of EQC based on BT.

- **Investigation of STW schemes**

The study delved into investigating the STW scheme for EQC. Prior to this research, there had been no exploration of the application of STW in the context of EQC. This theoretical contribution expands the knowledge base in the field of educational assessment by introducing and examining the STW scheme approach.

- **Exploration of ML and DL in limited dataset settings**

This research makes a theoretical contribution by addressing a previously unexplored research gap: the potential superiority of ML over DL in situations characterized by limited dataset sizes. The study conducted a comprehensive comparative analysis of ML and DL models to investigate and provide insights into this specific research question.

5.2.2 Practical Contributions

- **A novel term weighting model**

  The primary practical contribution of this study is introducing an enhanced term weighting model. By proposing and validating this model, the study provides a practical solution to improve the accuracy of EQC based on the BT.

- **Creation of a comprehensive dataset**

  The study's practical contribution includes developing and publicly releasing a substantial dataset comprising more than 700 labelled examination questions. This dataset is the largest publicly available dataset for researchers and practitioners in the EQC domain.

5.3 Limitations

This research presents multiple limitations that shape its scope and applicability. Firstly, it focuses exclusively on open-ended questions, omitting close-ended questions and coding-related programming questions, thus narrowing its relevance to a specific question type and limiting its broader application. Secondly, the study faces the common limitation of dataset size, as it acknowledges the absence of an extensive benchmark dataset for question classification, potentially hindering the generalization of findings.

5.4 Future Work

This study has yielded several valuable insights that pave the way for future research directions. One of the outcomes of this study is the positive impact and contribution of the ICF portion of the TF-ICF, the STW scheme. In

the future, the hybridisation of the ICF portion of TF-ICF with the proposed ETFPOS-IDF with IDF and without IDF incorporation to investigate whether combining them has a positive impact on the performance of classification. Another future research can be finding a method to calculate the optimal weight difference between the BT action and supporting verbs automatically rather than manually tuning. This exploration could offer valuable insights into fine-tuning the weighting scheme for more effective classification. In the context of limited data, the outcome of this study demonstrated that the ML with domain-specific term weighting can outperform the DL models. Given the current constraints of labelled data for EQC, prioritising ML approaches over DL models is advisable and recommended, as the latter typically requires a substantial amount of data to achieve satisfactory performance. The study also recommends building a larger labelled examination question dataset, which can be used to validate the proposed solution further. Furthermore, leveraging ontologies to improve the accuracy and interpretability of classification can be explored in the future. Moreover, it is worth exploring large language models like GPT and LLaMA for their potential in zero and few-shot classification tasks.

5.4 Concluding Remarks

In conclusion, this study has successfully investigated the impact of discriminating verb types in EQC, introducing the novel ETFPOS-IDF term weighting model. It has revealed the significance of considering verb types and has provided valuable insights into the superiority of domain-specific term weighting over DL models in limited data contexts. These findings contribute to the field of EQC and suggest promising avenues for future research.

REFERENCES

[1] S. Yilmaz and S. Toklu, "A deep learning analysis on question classification task using Word2vec representations," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2909–2928, 2020, doi: 10.1007/s00521-020-04725-w.

[2] S. R. Duggenpudi, S. Varma, and R. Mamidi, "Samvaadhana: A Telugu dialogue system in hospital domain," in *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, 2019, pp. 234–242. doi: 10.18653/v1/d19-6126.

[3] T. Gupta and E. Kumar, "Learning Improved Class Vector for Multi-Class Question Type Classification," in *Proceedings of the 3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC 2021)*, 2021, pp. 113–121. doi: 10.2991/ahis.k.210913.015.

[4] H. Sharma, R. Mathur, T. Chintala, S. Dhanalakshmi, and R. Senthil, "An effective deep learning pipeline for improved question classification into bloom's taxonomy's domains," *Educ. Inf. Technol.*, pp. 1–41, Oct. 2022, doi: https://doi.org/10.1007/s10639-022-11356-2.

[5] S. Shaikh, S. M. Daudpotta, and A. S. Imran, "Bloom's Learning Outcomes' Automatic Classification Using LSTM and Pretrained Word Embeddings," *IEEE Access*, vol. 9, pp. 117887–117909, 2021, doi: 10.1109/ACCESS.2021.3106443.

[6] N. Yusof and C. J. Hui, "Determination of Bloom's cognitive level of question items using artificial neural network," in *10th International Conference on Intelligent Systems Design and Applications*, 2010, pp. 866–870. doi: 10.1109/ISDA.2010.5687152.

[7] M. Mohammed and N. Omar, "Question classification based on Bloom's Taxonomy using enhanced TF-IDF," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, no. 4–2, pp. 1679–1685, 2018, doi: 10.18517/ijaseit.8.4-2.6835.

[8] A. Aninditya, M. A. Hasibuan, and E. Sutoyo, “Text Mining Approach Using TF-IDF and Naive Bayes for Classification of Exam Questions Based on Cognitive Level of Bloom’s Taxonomy,” in *2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, 2019, pp. 112–117. doi: 10.1109/IoTaIS47347.2019.8980428.

[9] D. A. Abduljabbar and N. Omar, “Exam questions classification based on Bloom’s taxonomy cognitive level using classifiers combination,” *J. Theor. Appl. Inf. Technol.*, vol. 78, no. 3, pp. 447–455, 2015.

[10] A. A. Yahya, Z. Toukal, and A. Osman, “Bloom’s Taxonomy–Based Classification for Item Bank Questions Using Support Vector Machines,” in *Modern Advances in Intelligent Systems and Tools*, vol. 431, 2012, pp. 135–140.

[11] A. Osman and A. A. Yahya, “Classifications of Exam Questions Using Linguistically-Motivated Features: A Case Study Based on Bloom’s Taxonomy,” in *The Sixth International Arab Conference on Quality Assurance in Higher Education*, 2016, vol. 2016.

[12] A. Sangodiah, R. Ahmad, and W. F. W. Ahmad, “Taxonomy based features in question classification using support vector machine,” *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 12, pp. 2814–2823, 2017.

[13] M. Mohammed and N. Omar, “Question classification based on Bloom’s taxonomy cognitive domain using modified TF-IDF and word2vec,” *PLoS One*, vol. 15, no. 3, pp. 1–21, 2020, doi: 10.1371/journal.pone.0230442.

[14] A. Alsaeedi, “A survey of term weighting schemes for text Classification,” *Int. J. Data Mining, Model. Manag.*, vol. 12, no. 2, pp. 237–254, 2020, doi: 10.1504/IJDMMM.2020.106741.

[15] R. B. Sachin and M. S. Vijay, “A survey and future vision of data mining in educational field,” in *Proceedings - 2012 2nd International Conference on Advanced Computing and Communication Technologies, ACCT 2012*, 2012, pp. 96–100. doi: 10.1109/ACCT.2012.14.

[16] S. H. Liao, P. H. Chu, and P. Y. Hsiao, “Data mining techniques and applications - A decade review from 2000 to 2011,” *Expert Syst. Appl.*, vol. 39, no. 12, pp. 11303–11311, 2012, doi: 10.1016/j.eswa.2012.02.063.

[17] C. Valois Batista, C. Valois, B. Marcius, and A. De Oliveira, “Recommender systems in social networks,” *JISTEM - J. Inf. Syst. Technol. Manag.*, vol. 8, no. 3, pp. 681–716, Dec. 2011, doi: 10.4301/S1807-17752011000300009.

[18] R. A. Huebner, “A Survey of Educational Data-Mining Research,” *Res. High. Educ. J.*, vol. 9, 2013, doi: 10.1038/1941006a0.

[19] F. Amanto, V. Moscato, A. Picariello, and G. Sperl′, “Recommender Systems and Social Networks: an application in Cultural Heritage,” *J. Vis. Lang. Sentient Syst.*, vol. 2, 2016.

[20] R. Jindal and M. D. Borah, “A Survey on Educational Data Mining and Research Trends,” *Int. J. Database Manag. Syst.*, vol. 5, no. 3, pp. 53–73, 2013, doi: 10.5121/ijdms.2013.5304.

[21] S. K. Mohamad and Z. Tasir, “Educational Data Mining: A Review,” *Procedia - Soc. Behav. Sci.*, vol. 97, pp. 320–324, 2013, doi: 10.1016/j.sbspro.2013.10.240.

[22] “educationaldatamining.org.” https://educationaldatamining.org/ (accessed Oct. 22, 2022).

[23] C. Romero and S. Ventura, “Educational data mining: A review of the state of the art,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 40, no. 6, pp. 601–618, 2010, doi: 10.1109/TSMCC.2010.2053532.

[24] M. Vranić, D. Pintar, and Z. Skočir, “The use of data mining in education environment,” in *Proceedings of the 9th International Conference on Telecommunications, ConTEL 2007*, 2007, pp. 243–250. doi: 10.1109/CONTEL.2007.381878.

[25] R. S. J. D. Baker and K. Yacef, “The State of Educational Data Mining in 2009 : A Review and Future Visions,” *J. Educ. Data Min.*, vol. 1, no. 1, pp. 3–16, 2009.

[26] C. Romero and S. Ventura, “Educational data mining: A survey from 1995 to 2005,” *Expert Syst. Appl.*, vol. 33, no. 1, pp. 135–146, 2007, doi: 10.1016/j.eswa.2006.04.005.

[27] Z. Xu, H. Yuan, and Q. Liu, “Student Performance Prediction Based on Blended Learning,” *IEEE Trans. Educ.*, vol. 64, no. 1, pp. 66–73, 2021, doi: 10.1109/TE.2020.3008751.

[28] M. Muñoz-Organero, P. J. Muñoz-Merino, and C. D. Kloos, “Student behavior and interaction patterns with an lms as motivation predictors in e-learning settings,” *IEEE Trans. Educ.*, vol. 53, no. 3, pp. 463–470, 2010, doi: 10.1109/TE.2009.2027433.

[29] F. Alshareef, H. Alhakami, T. Alsubait, and A. Baz, “Educational data mining applications and techniques,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 4, pp. 729–734, 2020, doi: 10.14569/IJACSA.2020.0110494.

[30] K. A. Osadi, M. Fernando, and W. V Welgama, “Ensemble Classifier based Approach for Classification of Examination Questions into Bloom’s Taxonomy Cognitive Levels,” *Int. J. Comput. Appl.*, vol. 162, no. 4, pp. 1–6, 2017.

[31] A. B. Zoric, “Benefits of Educational Data Mining,” *J. Int. Bus. Res. Mark.*, vol. 6, no. 1, pp. 12–16, 2020, doi: 10.18775/JIBRM.1849-8558.2015.61.3002.

[32] N. G. Ali and D. S. Hammad, “Classifying Exam Questions Based on Bloom’s Taxonomy using Machine Learning Approach,” in *Technologies for the Development of Information Systems TRIS-2019*, 2020, no. March, pp. 260–269. [Online]. Available: https://www.researchgate.net/profile/Nidaa-Ghalib/publication/339775553_CLASSIFYING_EXAM_QUESTIONS_BASED_ON_BLOOM’S_TAXONOMY_USING_MACHINE_LEARNING_APPROACH/links/5e650097299bf1744f68978c/CLASSIFYING-EXAM-QUESTIONS-BASED-ON-BLOOMS-TAXONOMY-USING-MACHINE-L

[33] A. A. Yahya, A. Osman, A. Taleb, and A. A. Alattab, “Analyzing the Cognitive Level of Classroom Questions Using Machine Learning Techniques,” *Procedia - Soc. Behav. Sci.*, vol. 97, pp. 587–595, 2013, doi: 10.1016/j.sbspro.2013.10.277.

[34] M. K. Taqi and R. Ali, “Automatic question classification models for computer programming examination: A systematic literature review,” *J. Theor. Appl. Inf. Technol.*, vol. 93, no. 2, pp. 360–374, 2016.

[35] K. Makhlouf, L. Amouri, N. Chaabane, and N. El-Haggar, “Exam Questions Classification Based on Bloom’s Taxonomy: Approaches and Techniques,” 2020. doi: 10.1109/ICCIS49240.2020.9257698.

[36] A. N. Darwazeh and R. M. Branch, “A Revision to The Revised Bloom’s Taxonomy,” in *2015 Annual Proceedings-Indianapolis*, 2015, vol. 2, no. 19, pp. 220–225.

[37] A. A. Yahya and A. Osman, “Automatic Classification of Questions into Bloom’s Cognitive Levels using Support Vector Machines,” in *The International Arab Conference on Information Technology*, 2011, pp. 1–6.

[38] B. S. Bloom, “Taxonomy of educational objectives : the classification of educational goals,” New York, NY, USA: David McKay Company, 1956.

[39] M. T. Chandio, S. M. Pandhiani, and R. Iqbal, “Bloom’s taxonomy: Improving assessment and teaching-learning process,” *J. Educ. Educ. Dev.*, vol. 3, no. 2, 2016.

[40] L. W. Anderson and D. R. Krathwohl, *A taxonomy for learning, teaching, and assessing: A revision of Bloom’s taxonomy of educational objectives*. Longman, 2001.

[41] T. T. Goh, N. A. A. Jamaludin, H. Mohamed, M. N. Ismail, and H. Chua, “Semantic Similarity Analysis for Examination Questions Classification Using WordNet,” *Appl. Sci.*, vol. 13, no. 14, p. 8323, 2023, doi: 10.3390/app13148323.

[42] A. Valentine and E. Oliveira, “Creating a software application to help university educators to reflect on the cognitive complexity of their exam questions, using Bloom’s Taxonomy and automated classification,” *ASCILITE Publ.*, pp. 568–572, Nov. 2023, doi: 10.14742/apubs.2023.613.

[43] Y. Li, M. Rakovic, B. Xin Poh, D. Gasevic, and G. Chen, “Automatic Classification of Learning Objectives Based on Bloom’s Taxonomy,” in *Proceedings of the 15th International Conference on Educational Data Mining*, 2022, no. July, pp. 530–537. doi: https://doi.org/10.5281/zenodo.6853191.

[44] Z. Hui, J. Liu, and L. Ouyang, “Question classification based on an extended class sequential rule model,” in *Proceedings of 5th International Joint Conference on Natural Language Processing*, 2011, pp. 938–946. Accessed: Oct. 24, 2022. [Online]. Available: https://aclanthology.org/I11-1105

[45] S. F. Kusuma, D. Siahaan, and U. L. Yuhana, “Automatic Indonesia’s questions classification based on bloom’s taxonomy using Natural Language Processing a preliminary study,” 2016. doi: 10.1109/ICITSI.2015.7437696.

[46] W. C. Chang and M. S. Chung, “Automatic applying Bloom’s taxonomy to classify and analysis the cognition level of english question items,” in *2009 Joint Conferences on Pervasive Computing, JCPC 2009*, 2009, pp. 727–734. doi: 10.1109/JCPC.2009.5420087.

[47] N. Haris, Syahidah Sufi and Omar, “A Rule- based Approach in Bloom ’ s Taxonomy Question Classification through Natural Language Processing,” in *2012 7th international conference on computing and convergence technology (ICCCT)*, 2012, pp. 410–414.

[48] S. Das, S. K. Das Mandal, and A. Basu, “Identification of cognitive learning complexity of assessment questions using multi-class text classification,” *Contemp. Educ. Technol.*, vol. 12, no. 2, pp. 1–14, 2020, doi: 10.30935/cedtech/8341.

[49] J. Zhang, C. Wong, N. Giacaman, and A. Luxton-Reilly, “Automated Classification of Computing Education Questions using Bloom’s Taxonomy,” in *Proceedings of the 23rd Australasian Computing Education Conference*, 2021, pp. 58–65. doi: 10.1145/3441636.3442305.

[50] S. K. Patil and M. M. Shreyas, “A Comparative Study of Question Bank Classification based on Revised Bloom’s Taxonomy using SVM and K-NN,” 2018. doi: 10.1109/ICECIT.2017.8453305.

[51] A. Sangodiah, R. Ahmad, and W. F. W. Ahmad, “A review in feature extraction approach in question classification using Support Vector Machine,” in *Proceedings IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2014, no. November, pp. 536–541. doi: 10.1109/ICCSCE.2014.7072776.

[52] N. Omar *et al.*, “Automated Analysis of Exam Questions According to Bloom’s Taxonomy,” *Procedia - Soc. Behav. Sci.*, vol. 59, no. 1956, pp. 297–303, 2012, doi: 10.1016/j.sbspro.2012.09.278.

[53] K. Jayakodi, M. Bandara, and I. Perera, “An automatic classifier for exam questions in Engineering: A process for Bloom’s taxonomy,” in *Proceedings of 2015 IEEE International Conference on Teaching, Assessment and Learning for Engineering(TALE)*, Jan. 2015, pp. 195–202. doi: 10.1109/TALE.2015.7386043.

[54] K. Jayakodi, M. Bandara, I. Perera, and D. Meedeniya, “WordNet and Cosine Similarity based Classifier of Exam Questions using Bloom’s Taxonomy,” *Int. J. Emerg. Technol. Learn.*, vol. 11, no. 04, pp. 142–149, Apr. 2016, doi: 10.3991/ijet.v11i04.5654.

[55] A. Sangodiah, T. J. San, Y. T. Fui, L. E. Heng, R. K. Ayyasamy, and N. Binti A Jalil, “Identifying Optimal Baseline Variant of Unsupervised Term Weighting in Question Classification Based on Bloom Taxonomy,” *MENDEL*, vol. 28, no. 1, pp. 8–22, 2022.

[56] M. D. Laddha, V. T. Lokare, A. W. Kiwelekar, and L. D. Netak, “Classifications of the summative assessment for revised bloom’s taxonomy by using deep learning,” *Int. J. Eng. Trends Technol.*, vol. 69, no. 3, pp. 211–218, 2021, doi: 10.14445/22315381/IJETT-V69I3P232.

[57] F. Baharuddin and M. F. Naufal, “Fine-Tuning IndoBERT for Indonesian Exam Question Classification Based on Bloom’s Taxonomy,” *J. Inf. Syst. Eng. Bus. Intell.*, vol. 9, no. 2, pp. 253–263, 2023, doi: 10.20473/jisebi.9.2.253-263.

[58] F. Debole and F. Sebastiani, “Supervised term weighting for automated text categorization,” in *Proceedings of the 2003 ACM symposium on Applied computing-SAC ’03*, 2003, pp. 784–788. doi: 10.1145/952532.952688.

[59] Y. Gu and X. Gu, “A supervised term weighting scheme for multi-class text categorization,” in *Intelligent Computing Methodologies*, 2017, vol. 10363, pp. 436–447. doi: 10.1007/978-3-319-63315-2_38.

[60] X. Quan, W. Liu, S. Member, and B. Qiu, “Term Weighting Schemes for Question Categorization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 1009–1021, 2011.

[61] D. Wang and H. Zhang, “Inverse-Category-Frequency based Supervised Term Weighting Schemes for Text Categorization,” *J. Inf. Sci. Eng.*, vol. 29, pp. 209–225, 2013.

[62] F. Ren and M. G. Sohrab, “Class-indexing-based term weighting for automatic text classification,” *Inf. Sci. (Ny).*, vol. 236, pp. 109–125, 2013, doi: 10.1016/j.ins.2013.02.029.

[63] G. Domeniconi, G. Moro, R. Pasolini, and C. Sartori, “A study on term weighting for text categorization: A novel supervised variant of tf.idf,” in *Proceedings of 4th International Conference on Data Management Technologies and Applications (DATA-2015)*, 2015, pp. 26–37. doi: 10.5220/0005511900260037.

[64] K. Chen, Z. Zhang, J. Long, and H. Zhang, “Turning from TF-IDF to TF-IGM for term weighting in text classification,” *Expert Syst. Appl.*, vol. 66, pp. 245–260, Dec. 2016, doi: 10.1016/j.eswa.2016.09.009.

[65] T. Dogan and A. K. Uysal, “Improved inverse gravity moment term weighting for text classification,” *Expert Syst. Appl.*, vol. 130, pp. 45–59, 2019, doi: 10.1016/j.eswa.2019.04.015.

[66] L. Chen, L. Jiang, and C. Li, “Modified DFS-based term weighting scheme for text classification,” *Expert Syst. Appl.*, vol. 168, p. 114438, 2021, doi: 10.1016/j.eswa.2020.114438.

[67] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: Analyzing Text with the Natural Language Toolkit*. Sebastopol, CA, USA: O’Reilly Media, Inc., 2009.

[68] S. Loria, “textblob Documentation,” *Release 0.15*, vol. 2, 2018.

[69] “The Stanford Natural Language Processing Group.” https://nlp.stanford.edu/software/tagger.shtml (accessed Aug. 15, 2022).

[70] M. Honnibal and I. Montani, “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing,” 2017.

[71] A. Sangodiah, Y. T. Fui, L. E. Heng, N. A. Jalil, R. K. Ayyasamy, and K. H. Meian, “A Comparative Analysis on Term Weighting in Exam Question Classification,” in *5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2021, pp. 199–206. doi: 10.1109/ISMSIT52890.2021.9604639.

[72] C. Cortes, V. Vapnik, and L. Saitta, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[73] B. Taha, “What is SVM | Build an Image Classifier With SVM.” https://www.analyticsvidhya.com/blog/2021/06/build-an-image-classifier-with-svm/ (accessed Feb. 15, 2023).

[74] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[75] A. Bouaziz, C. Dartigues-Pallez, C. da Costa Pereira, F. Precioso, and P. Lloret, “Short Text Classification Using Semantic Random Forest,” in *International Conference on Data Warehousing and Knowledge Discover*, 2014, pp. 288–299. doi: 10.1007/978-3-319-10160-6_26.

[76] X. Luo, “A New Text Classifier Based on Random Forests,” in *2nd International Conference on Materials Engineering and Information Technology Applications (MEITA 2016)*, 2017, pp. 290–293. doi: 10.2991/meita-16.2017.60.

[77] D. P. Kavadi, P. Ravikumar, and K. Srinivasa Rao, “A new supervised term weight measure for text classification,” *Int. J. Adv. Sci. Technol.*, vol. 29, no. 6, pp. 3115–3128, 2020.

[78] Y. Liu, Y. Wang, and J. Zhang, “New machine learning algorithm: Random forest,” in *International Conference on Information Computing and Applications*, 2012, pp. 246–252. doi: 10.1007/978-3-642-34062-8_32.

[79] A. Chauhan, “Random Forest Classifier and its Hyperparameters.” https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6 (accessed Feb. 15, 2023).

[80] R. F. de Mello, L. J. Senger, and L. T. Yang, “Automatic text classification using an artificial neural network,” *IFIP Adv. Inf. Commun. Technol.*, vol. 172, pp. 215–238, 2005, doi: 10.1007/0-387-24049-7_12/COVER.

[81] P. Lakshmi Prasanna and D. Rajeswara Rao, “Text classification using artificial neural networks,” *Int. J. Eng. Technol.*, vol. 7, no. 1.1 Special Issue 1, pp. 603–606, 2018, doi: 10.14419/IJET.V7I1.1.10785.

[82] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Jan. 2011.

[83] R. D. King, O. I. Orhobor, and C. C. Taylor, “Cross-validation is safe to use,” *Nat. Mach. Intell.*, vol. 3, no. 4, p. 276, 2021, doi: 10.1038/s42256-021-00332-z.

[84] P. Refaeilzadeh, L. Tang, and H. Liu, “Cross-Validation,” in *Encyclopedia of Database Systems*, Boston, MA, USA: Springer, 2009, pp. 532–538. doi: 10.1007/978-0-387-39940-9_565.

[85] M. Hossin and M. N. Sulaiman, “A Review on Evaluation Metrics for Data Classification Evaluations,” *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, pp. 01–11, 2015, doi: 10.5121/ijdkp.2015.5201.

[86] M. Grandini, E. Bagli, and G. Visani, “Metrics for multi-class classification: an overview,” *arXiv Prepr. arXiv2008.05756*, 2020.