IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 4, 2021, accepted August 15, 2021, date of publication August 20, 2021, date of current version August 31, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3106443

Bloom's Learning Outcomes' Automatic Classification Using LSTM and Pretrained Word Embeddings

SARANG SHAIKH[1,2], SHER MUHAMMAD DAUDPOTTA[1], AND ALI SHARIQ IMRAN[3], (Member, IEEE)

[1]Department of Computer Science, Sukkur IBA University, Sukkur 65200, Pakistan
[2]Department of Information Security and Communication Technology (IIK), Norwegian University of Science and Technology (NTNU), 2815 Gjøvik, Norway
[3]Department of Computer Science, Norwegian University of Science and Technology (NTNU), 2815 Gjøvik, Norway

Corresponding author: Ali Shariq Imran (ali.imran@ntnu.no)

This work was supported by the Department of Computer Science (IDI), Faculty of Information Technology and Electrical Engineering, Norwegian University of Science & Technology (NTNU), Gjøvik, Norway.

ABSTRACT Bloom's taxonomy is a popular model to classify educational learning objectives into different learning levels for three domains including cognitive, affective and psycho motor. Each domain is further detailed into different levels. The cognitive domain includes knowledge, comprehension, application, analysis, synthesis and evaluation levels. In educational institutions, designing course learning outcomes (CLOs) as per different levels of Bloom and mapping of assessment items on designed CLOs is an important task — every semester, faculty and administrators read thousands of statements to complete the tedious task of such mapping of CLOs and assessment items into Bloom's levels for an improved student learning. This paper proposes LSTM based deep learning model to perform classification of CLOs and assessment items in different levels of Bloom in cognitive domain. Although, there has been some attempts in the literature to automatically assign Bloom's taxonomy category using keywords-based approach but it suffers from the problem of low accuracy and overlapping of keywords. Initially, when we performed keywords-based approach on our datasets we achieved an overall accuracy of 55% for classification of CLOs and assessment items into Bloom's taxonomy. The proposed model predicts Bloom's level for CLO and assessment question item, respectively. The proposed model is simple in terms of the architecture as compared to other deep learning models reported in literature and achieves classification accuracy of 87% and 74% on CLOs and assessment question items, respectively. The proposed model obtained 3% increase in overall accuracy comparing to an existing study for the same task. To the best of our knowledge, this is first attempt towards applying deep learning on classifying educational objectives in Bloom's levels.

INDEX TERMS Bloom's taxonomy, learning objectives, text classification, natural language processing (NLP), deep learning, machine learning.

I. INTRODUCTION

Thinking ability is considered as a heart for all learning activities, without which no one can learn [1]. Every educational institution always tends to evaluate this thinking process by teaching, understanding, quality assessment and evaluation to ensure maximum learning of the students. First, the teaching and understanding in this process is carried out by teachers by designing the teaching material and a set of some course learning outcomes (CLOs) focusing student's thinking ability [2]. Second, the quality assessment is done by accreditation bodies and regulatory organizations [3]. Finally, the evaluation is done by conducting written examination. The educational institutions including teachers and accreditation bodies need hierarchical levels to differentiate thinking behaviors for students during the learning process. This also helps to understand what teacher is communicating and what student is perceiving during the learning process [4]. In 1956, Benjamin Bloom and a group of educational psychologists developed a classification system of different thinking behaviors important in learning namely, ''Bloom's Taxonomy'' [5]. Moreover, Krathwohl *et al.* in [6] have defined this taxonomy as ''Taxonomy of Educational Objectives''.

Bloom's taxonomy classifies thinking behaviors into three different domains: First, cognitive (related to mental behaviors); second affective (related to emotional behaviors) and

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

finally psychomotor (related to physical behaviors). Among all three categories, the cognitive domain has got much attention due to its high applicability in educational institutions [7]. The cognitive domain is further divided into six different hierarchical level structure of different thinking behaviors / levels involved in student's learning process (See Figure 24 in appendix). At each level, this approach has different keywords / action verbs associated which differentiate these levels from each other. (See Figure 2). Later on, the cognitive level was revised by few other experts and the revised Bloom's taxonomy was proposed. The levels of this revised taxonomy are mentioned in Figure 1. For the rest of this research study, we will only refer to cognitive domain of the revised taxonomy whenever there is a discussion on Bloom's taxonomy.

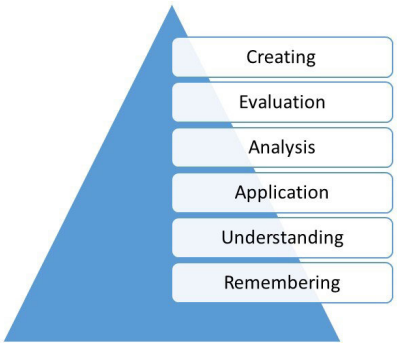

FIGURE 1. Revised bloom's taxonomy (Cognitive domain) hierarchy levels (Proposed by: Krathwohl *et al.* in [6].

Usually, the classification of course learning outcomes (CLOs) and questions on different levels of cognitive domain is done manually by teachers and accreditation bodies according to their own domain understanding. This is actually time consuming and often leads to mistakes due to human biasness. So there is a need to automate this process and this approach lies under emerging area of text classification. Indeed, there are some research works in the past that attempted to automate this process using keywords searching, natural language processing and machine learning techniques [8]–[15].

Previously, few studies [8]–[10] have employed keyword-based approaches to classify questions in Bloom's taxonomy levels. Though, the approach showed promising results, however it suffers from one major weakness of overlapping of several keywords in more than one Bloom's taxonomy cognitive levels [16]. (See Figure 2).

Consider the following example,

1) "Define the scope and importance of technical writing in academic and professional life"
2) "Define the basic principles and concepts as they relate to practical accounting problems"

Above, are the two CLO statements which belong to two different levels. CLO1 belongs to **understanding** level and CLO2 belongs to **remembering** level. However, if we analyze both CLOs we observed that both contains same action verb/keyword **"Define"**.

Consider another example,

1) "In your own words, how would you define transferable skills"
2) "Define compound interest"

Above, are the two questions which belong to two different classes. Q1 belongs to **understanding** level and Q2 belongs to **remembering** level. However, if we analyze both question statements we observed that both contains same action verb/keyword **"Define"**.

This is the major drawback of automatic keyword-based approaches for classification of CLOs and Questions. For human, to differentiate above CLOs or questions are easy to identify in their respective Bloom level but for machine using simple keyword-based approach it is erroneous. Therefore, the recent studies [8], [17]–[19] employed alternative machine learning-based approaches to classify the CLOs and questions. However, to the best of our knowledge, still these studies suffer from low accuracy [18]. This is because these studies employed existing conventional ML approaches. In addition, none of the existing proposed models are used in practice, due to the limited accuracy [13].

Hence, further improved automated text classification approaches are needed to improve the performance of existing studies in this domain. Recently, deep learning has shown very promising results as compared to traditional machine learning algorithms specially in the area of text classification [20]. Another problem that exists in this domain is the lack of already tagged datasets of CLOs and questions into cognitive levels. Most of researchers, working in this domain are developing their own datasets and those are publically available. Therefore, in this research project we have also developed our own dataset of CLOs classification into Bloom's taxonomy with the help of domain experts.

Our Research Contributions are:

1) The main contribution is an improved automatic Bloom's taxonomy CLOs and exam questions classification model utilizing the proposed deep LSTM model in combination with contextual domain embeddings.
2) Preparation of academic course CLOs and Questions' dataset, manually tagged by subject experts into one of six levels namely Remembering, Understanding, Application, Analysis, Evaluation and Creating.
3) The proposed approach addresses the issue of overlapping keywords in Bloom's taxonomy levels.
4) Although, the proposed improved classification model is the combination of existing techniques from the literature but it shows significant improvements to solve the given problem. The major contribution is towards the applied side of solving the problem of this research study.

The rest of the paper is organized as: Section II presents the literature review in the field of automatic classification into Bloom's taxonomy and introductory details about some

Level	Definition	Sample verbs					Sample behaviors
KNOWLEDGE	Student recalls or recognizes information, ideas, and principles in the approximate form in which they were learned.	arrange define describe duplicate	identify label list match	memorize name order outline	recognize relate recall repeat	reproduce select state	The student will define the 6 levels of Bloom's taxonomy of the cognitive domain.
COMPREHENSION	Student translates, comprehends, or interprets information based on prior learning.	explain summarize paraphrase describe illustrate classify	convert defend describe discuss distinguish estimate explain	express extend generalized give example(s) identify indicate	infer locate paraphrase predict Recognize	rewrite review select summarize translate	The student will explain the purpose of Bloom's taxonomy of the cognitive domain.
APPLICATION	Student selects, transfers, and uses data and principles to complete a problem or task with a minimum of direction.	use compute solve demonstrate apply construct	apply change choose compute demonstrate discover dramatize	employ illustrate interpret manipulate modify operate	practice predict prepare produce relate schedule	show sketch solve use write	The student will write an instructional objective for each level of Bloom's taxonomy.
ANALYSIS	Student distinguishes, classifies, and relates the assumptions, hypotheses, evidence, or structure of a statement or question	analyze categorize compare contrast separate apply	change discover choose compute demonstrate dramatize	employ illustrate interpret manipulate modify operate	practice predict prepare produce relate schedule	show sketch solve use write	The student will compare and contrast the cognitive and affective domains.
SYNTHESIS	Student originates, integrates, and combines ideas into a product, plan or proposal that is new to him or her.	create design hypothesize invent develop arrange assemble	categorize collect combine comply compose construct create	design develop devise explain formulate generate plan	prepare rearrange reconstruct relate reorganize revise	rewrite set up summarize synthesize tell write	The student will design a classification scheme for writing educational objectives that combines the cognitive, affective, and psychomotor domains.
EVALUATION	Student appraises, assesses, or critiques on a basis of specific standards and criteria.	Judge Recommend Critique Justify Appraise Argue	Assess Attach Choose Compare Conclude Contrast	Defend Describe Discriminate Estimate Evaluate Explain	Judge Justify Interpret Relate Predict	Rate Select Summarize Support Value	The student will judge the effectiveness of writing objectives using Bloom's taxonomy.

Reference: http://chiron.valdosta.edu/whuitt/col/cogsys/bloom.html

FIGURE 2. Bloom's taxonomy - cognitive domain (Action verbs).

recent approaches. Section III details the methodology used for proposed system, its construction, working and evaluation. Section IV, shows the experimental results of the proposed system in comparison with state-of-the-art models. Section V, highlights major insights from the experimental results. Section VI concludes the research study and Section VII presents the future work.

II. LITERATURE REVIEW

This section explains major background studies related to Bloom's taxonomy in general, existing approaches for automatic classification of CLOs and questions into Bloom's taxonomy and overview of relevant techniques including text classification and deep learning.

A. BLOOM'S TAXONOMY

Chang *et al.* in [2] explained that the Bloom's taxonomy (cognitive domain) is being actively used by educational institutions (i-e: teachers and accreditation bodies) to define course learning outcomes (CLOs), to design teaching material and to assess examination questions in order to find out problems in student's learning. However, Krathwohl *et al.* in [6] introduced revised version for Bloom's taxonomy - cognitive domain (See Figure 1). The major difference was, in revised version the Evaluation is on second highest level as compared to original taxonomy. Moreover, the new category Creating is on the top of revised taxonomy. The revised taxonomy was developed to show the intersection and different type of levels involved in learning. In this study, we have considered revised Bloom's taxonomy for the classification of CLOs and examination questions.

Recently, Swart *et al.* in [21] explained the importance of using Bloom's taxonomy for understanding and classification of CLOs. The authors observed that cognitive domain defines different thinking behaviors from simple memory recall to complex reasoning skills in terms of student's learning (See Figure. 24). They analyzed course learning outcomes (CLOs) of an Electronic Fundamental course offered in two universities in Romania and South Africa, respectively. The results from this study indicated that the first two levels of cognitive domain (remembering & comprehension) contributed overall 58% to the total CLOs. Meanwhile, the application and rest of the levels contributed 27% and 15% to the total CLOs, respectively. Also, Rahmatih *et al.* in [22],used the Bloom's taxonomy to analyze the student's questioning skills. The major take away from this study was that the questions were asked in the cognitive domain. Hence, it shows the important of cognitive domain in the education sector.

Moreover, Atiullah *et al.* in [23] used Bloom's taxonomy to evaluate the availability of higher order thinking skills in reading comprehension questions. The authors collected a total 158 reading comprehension questions from 15 texts of English textbook for Grade X. The authors classified complete questions into Bloom taxonomy using manual intuitive approach. The results indicated that majority of these questions (i-e: 134) were categorized at remembering level and only 24 questions out of 158 were categorized at other higher levels (i-e: comprehension, application). It was concluded that English textbook of Grade X is lacking high order thinking capabilities as 85% of reading comprehension questions were below the comprehension level in the Bloom's taxonomy. Hence, the above studies and discussions have clearly explained the importance of using Bloom's taxonomy in education. Wijanarko *et al.* in [24], proposed the use of Bloom's taxonomy while generate questions from the unstructured content. The whole idea was to evaluate the generated questions into the Bloom's taxonomy levels, if it satisfies the learning outcomes.

B. TEXT CLASSIFICATION

Text classification is becoming much more important these days due to exponential growth in complex texts generated by Internet, which requires an in-depth understanding of machine learning algorithms that can automatically categorize texts into many real-world applications. Most of the breakthroughs in text classification are due to the efficiency of recent techniques in understanding complex relationships in text data [14]. Moreover, text classification has been actively used and studied in applications like Information Retrieval [15], [25], [26] and Information Filtering [27], [28].

C. AUTOMATIC CLASSIFICATION OF CLOs AND EXAMINATION QUESTIONS INTO BLOOM's TAXONOMY

Automatic classification of educational objectives into Bloom's taxonomy can be defined as a task of classifying CLOs or exam questions from course material or examination papers, respectively. In previous studies, several researchers have tried to solve this problem using automatic ML-based and NLP-based techniques. The literature highlights two major approaches in this domain namely, Keyword-Based [1], [2], [29] and Text-Classification-Based [13], [19], [30]–[33].

1) KEYWORD-BASED APPROACH

Bengio *et al.* in [17] applied keywords based approach by searching keywords for each level. (See Figure 2) in the Appendix for list of available keywords). The keywords were searched for different CLOs and Question statements for the sake of testing. Without the use of machine learning / text classification approach the authors managed to get an accuracy of 75% for only Remembering level. Moreover, for Evaluating level the authors got only 25% accuracy. The average accuracy for all 6 cognitive domain levels was just 47%. Omar *et al.* in [1] applied NLP-based technique to identify and use important keywords. After identification, a rule-based approach was used by authors for identification of the desired cognitive domain level. The authors applied this approach on 100 questions (70 training set and 30 test set items). The authors reported very low accuracy as the training set is not enough to learn the rules. In Keyword-Based approach, the accuracy is good up to two basic levels i-e: Remembering and Comprehension because the CLOs / Question statements are very simple and straight forward and there is no issue of keyword overlapping.

Although, all the researches above are done with the general keywords of Bloom's taxonomy (cognitive domain). But Christian *et al.* in [34], also prepared the list of new verbs for the Bloom's taxonomy. The new proposed verbs are 84 in total, out of which 34 are technical verbs. They haven't made the list public yet.

2) TEXT-CLASSIFICATION-BASED APPROACH

Zhang *et al.* in [19], applied machine learning algorithms to classify questions related to the computing education into the Bloom's taxonomy. The total questions were 504 and manually annotated by the education experts. The authors reported the highest accuracy of 82% on the test set where the ratio of training and test set was 90 and 10, respectively.

Manjushree *et al.* in [35] applied deep learning based models (CNN and LSTM) for the classification of assessment items into cognitive level of Bloom's taxonomy. The authors collected and manually labelled the dataset of total of 844 instances into six levels from the software engineering course. Furthermore, the authors used the train-test split ratio (70-30%) to assess the performance of CNN and LSTM models. The authors got the highest performance of 80% on the test set using CNN model. This is the most recent work done in this domain.

Mohammed Manal *et al.* in [36] applied three ML-based classifiers (KNN, Logistic Regression and SVM) with two feature engineering techniques (TF-IDF, Word2Vec) to classify questions into cognitive domain of the Bloom's taxonomy. The authors used two datasets. One was manually collected and labelled by them with 141 questions into six Bloom's taxonomy levels and second dataset was used from the literature by [31] with 600 questions divided into same six Bloom's taxonomy levels. The authors achieved satisfactory results on both datasets using train-test split approach. The average accuracy obtained for first dataset was 83.7% and for the second dataset they achieved 89.7%. This was a significant improvement in the results achieved by [31] on the same dataset although, the approach was promising but less generalizable and less scalable. As soon as, the amount of data will increase it will become a bottleneck for this approach.

Hoeij *et al.* in [30] applied ML-based SVM classifier for examination question classification into Bloom's taxonomy. They achieved more than 80% accuracy because the dataset was very small and almost all the questions written were according to keywords of Bloom's taxonomy cognitive

domain. The major drawback in this study was it is not necessary that always the examination questions contains these keywords. Osadi *et al.* in [13] combined multiple classifiers using ensemble approach for the question classification into Bloom's taxonomy. Since, their dataset was too small which contained only 100 questions so they proposed to further continue this approach on large dataset in the future work.

Yahya *et al.* in [31] developed a classification model to classify short essay questions on Bloom's taxonomy cognitive domain on two veterinary courses. The authors achieved an overall accuracy of 65%. Yusof *et al.* in [32] proposed different machine learning based methods to classify question into cognitive domain levels. The authors experimented with different feature engineering and ML models combinations and achieved highest accuracy of 76% with SVM classifier. Furthermore, Zhang *et al.* in [33] proposed a technique called Category Frequency-Inverse Document Frequency (CF-IDF). The proposed method used ANN (Artificial Neural Networks) to utilize the frequency of each class label in order to classify questions. The authors achieved only 60% accuracy for first three levels of cognitive domain.

D. DEEP LEARNING

Deep learning algorithms and architectures have done excellent advances in the fields like computer vision. Moreover, the recent support of deep learning for NLP can be observed from last 5 years articles indexed in WoS. (see Figure 3). In recent years, deep learning approaches depending on dense vector representations are producing promising results on variety of NLP tasks including text classification [20]. The major reason for this success is the use of word embeddings [37], [38] and deep neural network architectures [39]. Deep learning models supports automatic feature representations learning based on data as compared to traditional shallow machine learning models where the features representation learning is based on hand-crafted features. Collobert *et al.* in [40] shows that deep-learning models are leaving behind most of the traditional state-of-the-art approaches for text classification, named-entity-recognition (NER) and Part-of-speech (POS) tagging. Hence, the deep learning models are increasingly being used in NLP problems like machine translation, sentiment classification and text generation [18].

1) WORD EMBEDDINGS

This is the most often first data processing layer in deep learning models, which helps network to learn automatic feature representations learning by converting raw text into continuous real numbers. These word embeddings or dense vector representations work on simple hypothesis that words having similar meanings tends to occur in same context. The similarity between different word vectors is measured using cosine similarity [41]. Therefore, these embeddings are fast and efficient in order to capture context in most of the state-of-the-art core NLP tasks [42]–[44]. These

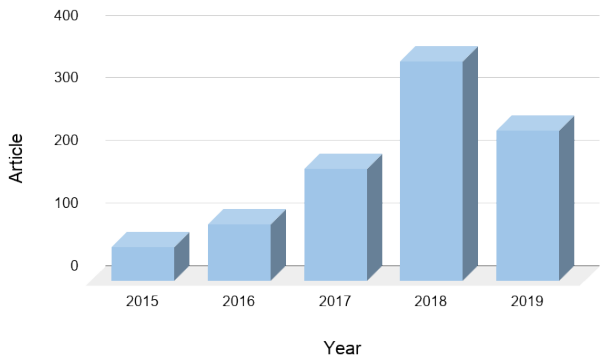

FIGURE 3. Deep learning based NLP papers indexed in WoS (Last 5 years).

representations are mainly learned through context in unsupervised manner [18].

The most efficient and actively used word embedding technique is word2vec, which comprised of two models namely continuous-bag-of-words (CBOW) and Skipgram models [45]. Bengio *et al.* in [17] explained that once these individual word representations are combined into sentence representation using joint probabilities for word sequences, make efficient representation for unseen sentences if the sentences are of same context because the network have already learned those representations. Moreover, the learning of word embedding can be done by using some pretrained embeddings like Glove [46], Elmo [47], BERT [48] and FastText [49]. These embeddings form the foundations for many current approaches in NLP using deep learning.

2) RECURRENT NEURAL NETWORKS

The term "recurrent" refers to perform same action/computation over sequence of data (i-e: sequence of tokens in text data). RNNs [50] is basically used for processing sequential information where each computation is performed over sequence of tokens and each next computation is dependent on previous computations and its results. In general, a fixed size vector is created to represent sequential information of tokens and in this way RNNs store information for previous computations to be used for current processing. In recent years, RNNs are being widely used in major NLP tasks namely language modeling [51], [52], machine translation [53], [54], speech recognition [55], [56] and image captioning [57]. RNN suffers from the problem of vanishing gradient [58] which makes it difficult to learn longer sequences.

To solve this problem, specifically for NLP problems different variants for RNN are used such as Long Short-Term Memory (LSTM) [59]. LSTM [60], [61] has additional "forget" gate as compared to simple RNN model. This mechanism helps it to overcome the problem of vanishing gradient as discussed above. This architecture consists of three gates namely, input, forget and output gates. The hidden state is calculated using combination of these three gates. Other variants of RNN are Gated Recurrent Units (GRUs) [62],

BiDirectional LSTM and GRUs [63]. This architecture of LSTM neural network is explained in detail in section III-G.

Previous studies discussed in Section II-C showed very good results in automatic classification into Blooms taxonomy. To the best of our knowledge, no study has used the recent word embeddings and deep learning models based approach for this problem. Hence, in this research we have adapted these recent approaches to perform automatic classification of CLOs and questions into Bloom's taxonomy.

III. METHODOLOGY

A. OVERVIEW

This section presents the overall research methodology which we have used to classify CLO or Question into six distinct classes namely, ''Remembering, Understanding, Application, Analysis, Evaluation and Creating''. Figure 4 depicts the overall research methodology.

The methodology is logically divided into two components namely, 1) Domain Understanding and Data Acquisition, 2) Construction of Proposed System. The details of each component is discussed in subsequent sections.

B. DOMAIN UNDERSTANDING

In order to understand more about the problem domain, we conducted interviews from various domain experts. The major purpose behind conducting these interviews was to know the different methods that are considered in various departments, like computer science, electrical engineering and business administration for categorization of CLOs and examination question into Bloom's taxonomy. The domain experts included coordinators from international accreditation bodies like Accreditation Board for Engineering and Technology (ABET) and Association to Advance Collegiate Schools of Business (AACSB), HoDs and subject specialist faculty members from different universities. The selection of experts was on the basis of their experience. Because, these are the peoples who are involved in the process of categorizing CLOs and examination questions and can explain different ways of doing this activity in a best manner. The total number of participants were 10, so we manually analyzed the responses of each question asked from the participants. The questions are mentioned in Appendix A. After reviewing all the interviews, we concluded three major points.

1) The categorization of CLOs and questions into Bloom's taxonomy is purely based on human understanding and is domain specific.
2) This is an important activity carried out in academic institutions, for the assessment of course quality as well as examination paper quality needed to quantify student's learning.
3) If a single CLO or question statement contains Bloom's keyword/action verb, which is overlapping on different levels then neighbouring words are checked to differentiate levels, as words are known through their company.

C. DATA ACQUISITION

As far as we know, there is no standard public data set available containing course learning outcomes (CLOs) and questions tagged into Bloom's taxonomy. For this study, a manually tagged data set of Sukkur IBA University is used. This will create a baseline for performing further experiments in this problem domain. Moreover, we have requested a dataset of questions categorized into Bloom's taxonomy from faculty members of Najran University, Saudi Arabia. We will use that dataset as a baseline as well to evaluate our proposed methodology because the authors in [31], [36] have also used this same data set for classification into Bloom's taxonomy. Usually, the faculty members create CLOs in course description documents at the start of the semester and assign those CLOs to questions asked in examination paper. The ABET or AACSB coordinator perform the mapping of CLOs into Bloom's taxonomy and ensure whether the mapping to Bloom's taxonomy is sufficient for maximum student's learning [64].

For this study, we have used two datasets. Table 1 depicts some of the important statistics for both datasets. For Dataset 1, a team from Department of Quality Enhancement Cell (QEC) was asked to manually tag the Bloom's taxonomy (cognitive domain) level to the compiled CLOs statements. The tagging was verified from faculty for related courses from three departments (i-e: computer science, electrical engineering and business administration) to minimize error. Figure 5 explains classwise distribution for Dataset 1. However, the Dataset 2 which we acquired from researchers of another existing study was already tagged into Bloom's taxonomy levels. Figure 6 shows classwise distributing of Dataset 2. We have used Dataset 1 to create baseline for our proposed system and Dataset 2 to evaluate our proposed system in comparison with same existing study on this dataset.

D. PROPOSED SYSTEM OVERVIEW

Figure 7 shows the abstract model of the proposed system. The proposed system accepts raw CLO / Question text at the input to classify it into one of the Bloom's taxonomy level in cognitive domain. The proposed system performs the following tasks,

1) The first step is a text pre-processing and cleaning that takes the input text and pre-process it by converting into lower case, removing stopwords and punctuation and converting all words to their root words using lemmatization.
2) Once the text is preprocessed, the next step is to compute numeric word vectors using skip-gram based word embedding in order to represent text into numeric features.
3) Finally, we use the Bloom's taxonomy level classifier to classify into one of the pre-defined categories.

Suppose, following raw CLO / question text is input,

''Draw the flow chart of the PNK System''

For this text, following triple would be generated,

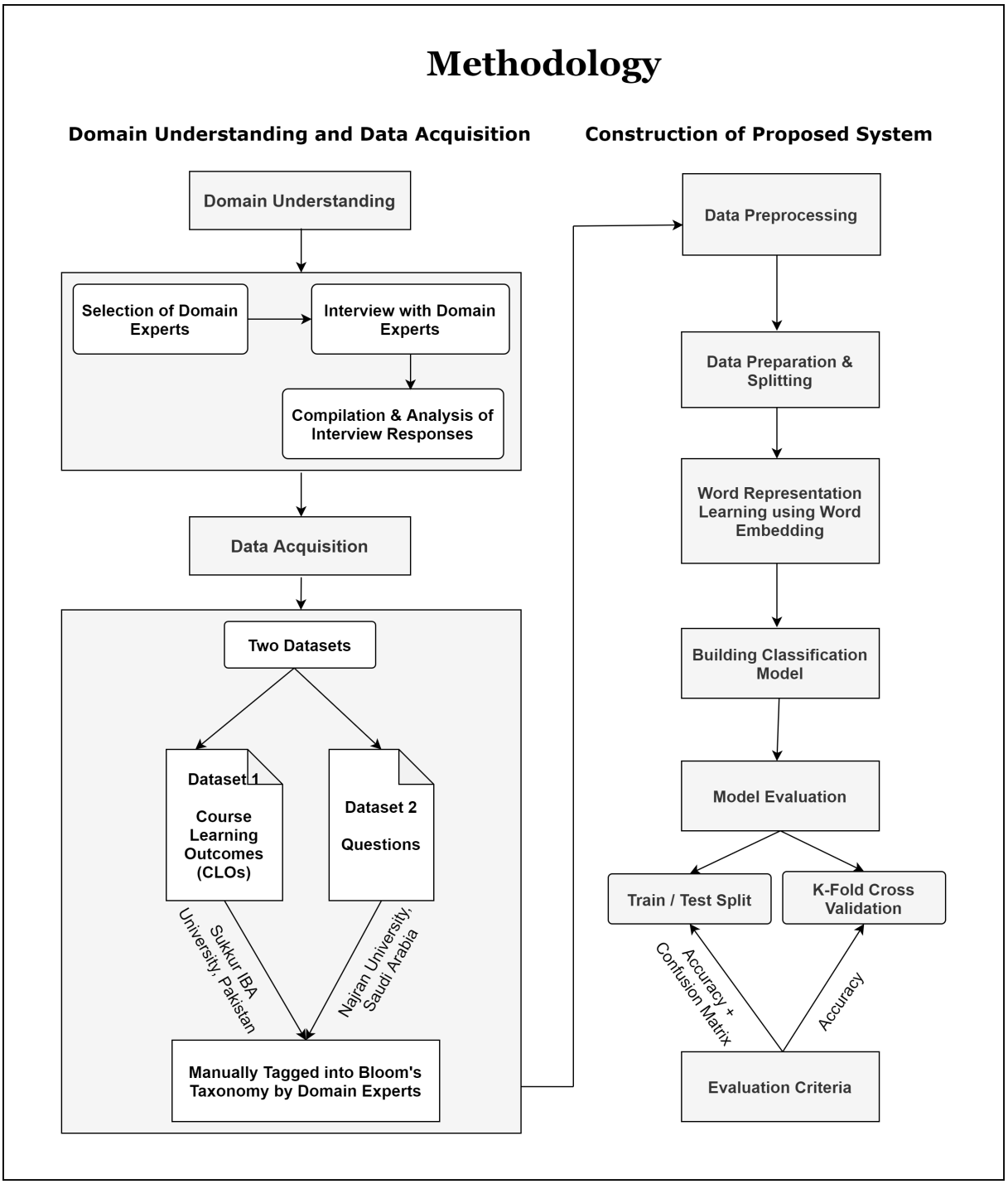

FIGURE 4. Methodology used for this research study.

(Prep, Embed, BLevel) = (draw flow chart pnk system, [0.2315,0.1224,-0.26339,....], Remembering)

where Prep = Preprocessed Text, Embed = Word Vectors and BLevel = Bloom's Taxonomy Level

Similarly, for following text,

"Explain the whole method of crushing"

Here, the triple would be,

(Prep, Embed, BLevel) = (explain whole method crushing, [−6.4062,6.7858,8.6112,...], Understanding)

In next few sections, we describe construction and working of each of these components of the proposed system.

E. CONSTRUCTION OF PROPOSED SYSTEM

This section presents construction of all the three modules involved in the proposed system. As shown in Figure 7, the proposed system is comprised of three modules data pre-processing and cleaning, Learning Word Representation using Word Embedding and Bloom's Taxonomy Level

TABLE 1. Statistics of datasets used for this study.

Title	Information	Source	Total Instances	Total Classes	Class wise Instances
Dataset 1	CLO statements from course documents	Sukkur IBA University	829	5	Remembering: **48** Understanding: **422** Application: **269** Analysis: **83** Evaluation: **11** Total: **829**
Dataset 2	Questions compiled and used in research study [31], [36]	Najran University, Saudi Arabia	600	6	Remembering: **100** Understanding: **100** Application: **100** Analysis: **100** Evaluation: **100** Creating: **100** Total: **600**

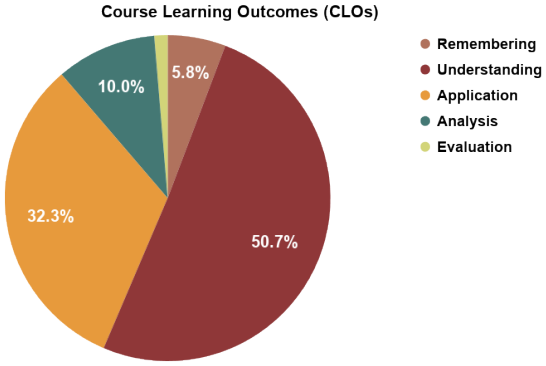

FIGURE 5. Classwise distribution (Dataset 1).

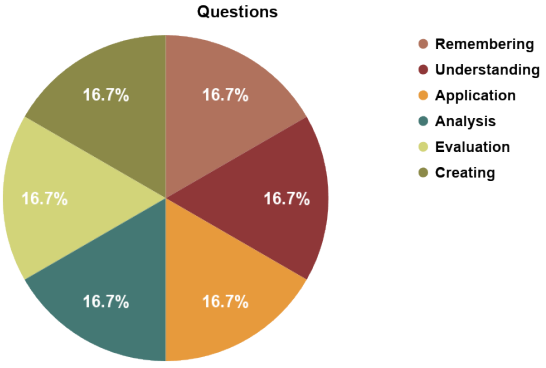

FIGURE 6. Classwise distribution (Dataset 2).

Classifier. The details of each module are discussed in detail in below sections.

1) DATA PRE-PROCESSING AND CLEANING

Several studies have shown that data pre-processing shows better classification results [45]. Therefore, in our collected datasets, we applied several pre-processing techniques to remove non-informative features from the data. In pre-processing, we converted the text data into lower case. In addition, we remove punctuation and stop words using regular expressions and pattern matching techniques. Besides this, we have also performed white space tokenization and wordnet lemmatization to preprocessed text. In tokenization, each question/CLO is converted into tokens or words, then words are converted to their root forms, such as offended to offend using wordnet lemmatizer. Algorithm 1 shows series of different pre-processing steps applied on raw datasets to get clean datasets as output.

Figure 8 and Figure 9 shows example of applying pre-processing steps to CLO and Questions data, respectively.

Algorithm 1 Data Pre-Processing and Cleaning

INPUT: Raw CLO / Question Text
OUTPUT: Preprocessed CLO / Question Text

```
sentences ← extractSentences(INPUT)
prepInput = Empty
while sent ∈ sentences do
    stSentence ← removeStpWords(sent)
    pSentence ← removePunc(stSentence)
    words ← extractWords(pSentence)
    prepSentence = Empty
    while word ∈ words do
        lword ← lower(word)
        lemmaWord ← WordNetLemmatizer(lword)
        prepSentence += lemmaword
        prepSentence += space
    end while
    prepInput.append(prepSentence)
end while
OUTPUT ← prepInput
```

2) DATA PREPARATION AND SPLITTING

After, doing data pre-processing and cleaning the next step in our proposed system is the preparation and splitting of data so that it can be used for model construction and its evaluation. Algorithm 2 explains series of steps which we performed on Dataset 1 and Dataset 2 to prepare training and test datasets for model construction and evaluation.

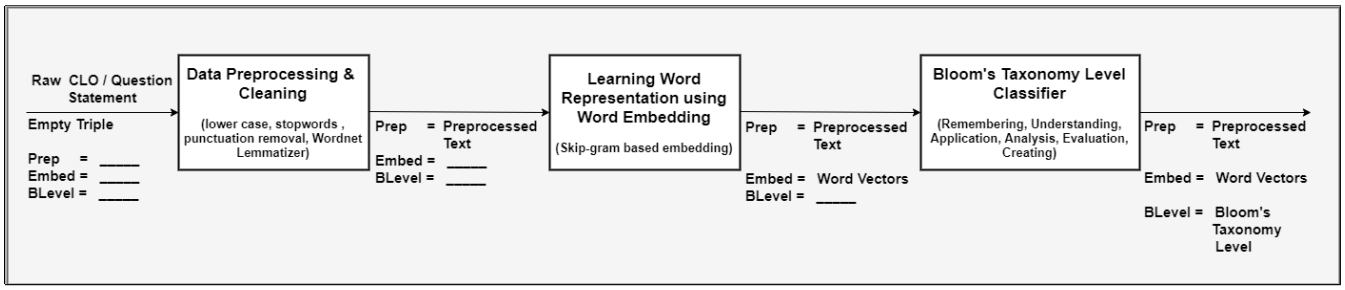

FIGURE 7. Proposed system overview.

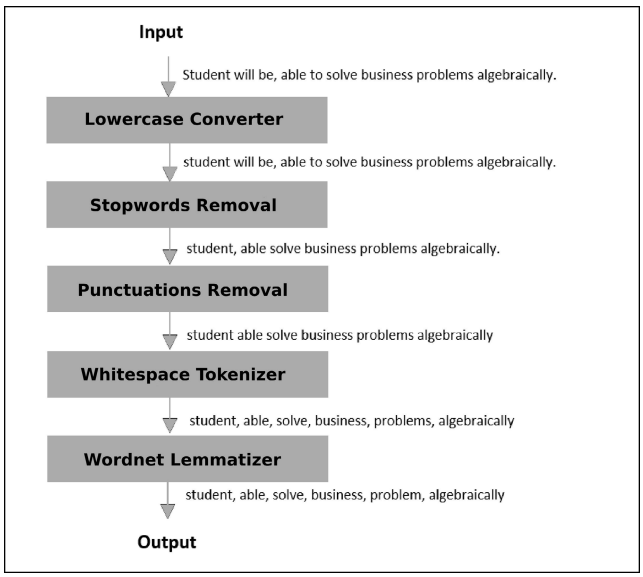

FIGURE 8. Example: Data pre-processing (Dataset 1).

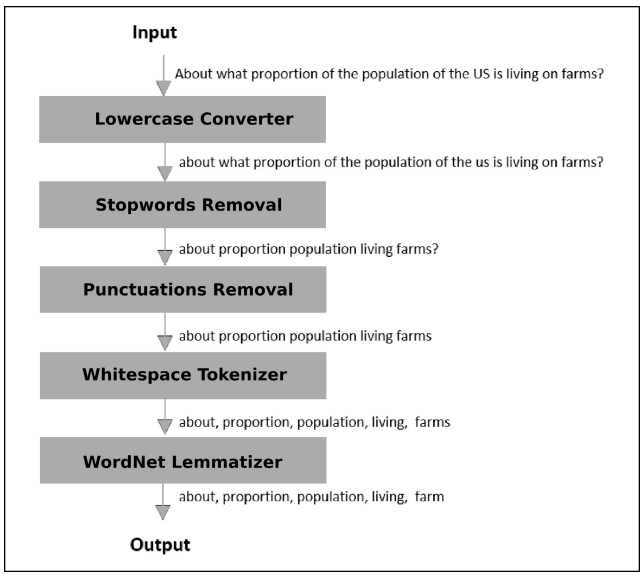

FIGURE 9. Example: Data pre-processing (Dataset 2).

Moreover, The selection of no. of unique words and maximum length is finalized after several experiments and are shown in later sections. (See Table 9 and Table 10) in the appendix section. Also, the selection of best test size to be used for the maximum performance is also selected after several experiments. (See Figure. 18).

Algorithm 2 Data Preparation and Splitting

INPUT: Preprocessed Input Text and Class Label
OUTPUT: Training and Test Data Partition

$uniqueWords \leftarrow uniqueWords$
$maxLength \leftarrow maxLen$
$testSize \leftarrow testSize$
$text \leftarrow$ sequences(Preprocessed Input Text)
$labels \leftarrow$ encoding(Class Label)
$data \leftarrow shuffling(text, labels)$
$trainData = partition(data, ratio = 1 - testSize)$
$testData = partition(data, ratio = testSize)$

3) LEARNING WORD REPRESENTATION USING WORD EMBEDDING

One of the major feature used in our proposed system is the semantic representation of words using the Word Embeddings. As both of our datasets are small, usually with small text datasets in deep learning, per-trained word embeddings are used [65]. Therefore, we decided to use pre-trained embeddings to learn efficient word representations for our datasets. We selected one of the recent pre-trained word embedding namely, ''Wiki Word Vectors''. These pre-trained embeddings were developed by Facebook AI Research in total 294 languages in which English is also included. Moreover, these embeddings were trained on Wikipedia text. The authors in [49] have explained these embeddings in detail. We will discuss these details in later section III-E4. We selected this embedding for our task because we expected that this will help us to get semantic similarities of words in a better way due to following reasons:

- This embedding is trained on Wikipedia text using technique of generating representation of a word based on its neighbouring words.
- Our datasets consist of maximum words from the Wikipedia corpus.

Additionally, we also experimented with the other pre-trained embeddings like Glove.6B.100D and GoogleNews-vectors-negative300 to evaluate which is better for our task. The performance comparison of these

pre-trained embeddings with our proposed pre-trained embedding will be discussed in later section.

4) PRE-TRAINED EMBEDDING "WIKI WORD VECTORS" FOR WORD REPRESENTATION

The pre-trained embedding used for this study was originally developed by a team of researchers from Facebook AI Research in 2017. The major motivation behind this development is based on research from neural network community where [66] proposed the use of feed-forward neural network to learn numeric representation of a word, based on occurrence of its left and right neighbouring words. This help the network to build understanding of words which are occurring with each other. The major issue in other pre-trained embeddings like Glove.6B.100D and GoogleNews-vectors-negative300 is, although these are continuous words representations which are trained on large corpus but these representations ignore the word morphology by using distinct vector to each word. This creates a limitation for rare or out-of-vocabulary words which were not the part of the training corpus. The model used to prepare "Wiki Word Vectors" pre-trained embeddings is an extension of original continuous skip gram model.

The initial skip gram model proposed by Mikolov [49] is defined as:

A dictionary for vocabulary of size K, where each word is identified using index k, defined in equation. 1.

$$k \in \{1, \ldots, K\} \tag{1}$$

The first assumption for skip-gram model is that a single word can be useful to generate its own surrounded neighbouring words inside a sequence of text. For Example, if we take below text sequence

"student will demonstrate basic proficiency in computer commonly used computer applications"

We take **"demonstrate"** as the middle target word by setting the words context window size = 2. From Figure 10, we can see that, the skip-gram network model is interested in calculating conditional probabilities for creating the context words, "student", "will", "basic" and "proficiency" that are around distance, not exceeding 2 words from the central target word "demonstrate".

But, there is not only single central target word for our consideration. For the given, text sequence each word is treated as target word and for each word we need to calculate conditional probabilities for its surrounded context words. So our input data becomes in below form of equation. 2

$$\{k \mid k \in K\} \left(\left[contextWords(k) \right], targetWord = k \right) \tag{2}$$

where K is the dictionary of words from training data, k is the individual word. According to equation 2 the text sequence becomes in the below form with context words window size = 2. ([will, demonstrate], student), ([student, demonstrate, basic], will), ([student, will, basic, proficiency], demonstrate), ([will, demonstrate, proficiency, in], basic), ([will, demonstrate, proficiency, in], basic), ([demonstrate, basic, in, computer], proficiency), …..

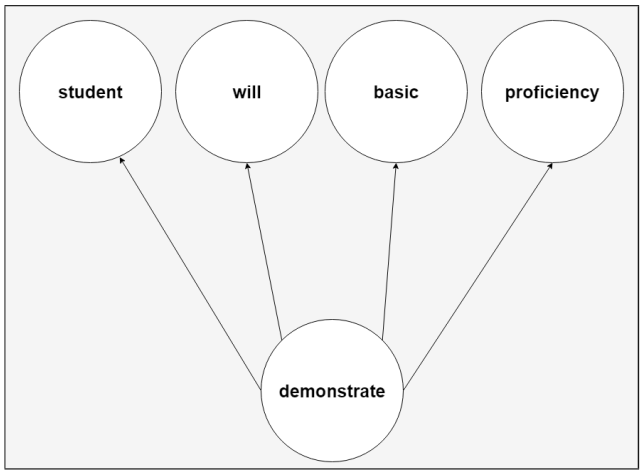

FIGURE 10. Skip-gram model context words example.

We know that, skip-gram network model tries to learn conditional probabilities of context words for a defined target word. Assuming that, all the context words are generated independent of each other. So, the skip-gram model is interested in calculating below.

$$P(\text{"}student\text{"}, \text{"}demonstrate\text{"}) \, . \, P(\text{"}will\text{"}, \text{"}demonstrate\text{"})$$
$$P(\text{"}basic\text{"}, \text{"}demonstrate\text{"}) \, . \, P(\text{"}proficiency\text{"}, \text{"}demonstrate\text{"})$$

Once, the target, context words tuples are formed for complete training data; the next step is to predict and learn word representations for context words with their respective target words using a neural network architecture. The neural network used for skip-gram model is a simple shallow neural architecture with three layers. 1) input layer, 2) single hidden layer and 3) output layer. The input layer is one-hot-encoded version of the input target word whereas, the output layer is the probability function of context words likely to appear with input target word. As, in neural network each successive layer is built by computing dot product of the layer with its weight matrix in addition of a bias using some non-linear activation function like logit, softmax. Therefore, we can define the skip-gram neural architecture as equation 3:

$$logit\left(w_k\right) = x_k W + b \tag{3}$$

where w_k and x_k are the target word, b and W are bias and weights matrices respectively. Also, $logit(w_k)$ return unnormalized scores at the output layer so we need to apply softmax activation at the output layer to normalize the probability scores defined as equation 4:

$$y_k = softmax\left(logit\left(w_k\right)\right) \tag{4}$$

The Softmax activation function is calculated using equation 5, where n is the total number of target, context words tuples.

$$Softmax\left(logit\left(w_k\right)\right) = \frac{\exp\left(logit\left(w_k\right)\right)}{\sum_{k=1}^{n} \exp\left(logit\left(w_k\right)\right)} \tag{5}$$

Figure 11 depicts the working for skip-gram neural network built for training example of below target, context words tuples representing context window $c = 2$.

$$P(\text{“}student''\text{, “}demonstrate''\text{) . }P(\text{“}will''\text{, “}demonstrate''\text{)}$$
$$P(\text{“}basic''\text{,“}demonstrate''\text{) .}P(\text{“}proficiency''\text{,“}demonstrate''\text{)}$$

First, we can see the input target word **"demonstrate"** is converted into one-hot-encoding vector of size of the vocabulary dictionary $v = 8$. Second, the weight matrix is created of size v x n, where n is the dimension for word embedding and this matrix is represented as p. This matrix represents each vocabulary word as single row and used as input into the hidden layer of size n. As, the $c = 2$, each training instance will feed forward 4 times to 4 output vectors (i-e: w_{k-2}, w_{k-1}, w_{k+1}, w_{k+2}). The hidden layer, linked to output layer with its weight matrix represented as p' with dimension of size n x v. Initially, the weight matrices p and p' are getting some random values inside neural network and the optimal values for representing these matrices are learned by optimizing the network using back propagation and minimizing the loss according to the equation. This will help the neural network to learn more meaningful word representations for representing context words with their respective target words.

F. LEARNING WORD REPRESENTATIONS USING PRE-TRAINED FOR DATASET1 AND DATASET 2

We computed word representations for our both datasets using pre-trained embedding explained in above section. We obtained embedding matrix of 300-dimension for all unique words in our datasets. This embedding matrix will serve as the weights for embedding layer in our proposed classification model which will help the proposed system to learn how different words are appeared with their different contexts.

Figure 12 shows the word embeddings scores of some words from our embedding matrix calculated as per above method.

Furthermore, Figure 13 shows the first 40 unique words according to their embedding matrix. This helps to understand how well these embeddings are computed. The words on the visual which are nearer to each other are the neighbouring words occurring with each other in same context and represents specific Bloom's taxonomy level. However, the different word groups away from each other are representation of different levels of Bloom's taxonomy.

For example, the words **"problem, describe, state, data"** mostly occurs with each other in neighbours. These words have the same context and represents the **"Remembering"** level. However, the words **"following, compare"** occurs with each other in neighbours. Therefore, these two have the same context but different context from previous words. So, they represents the **"Analysis"** level which is different from the level represented by previous words.

G. CONSTRUCTION OF BLOOM's TAXONOMY LEVEL CLASSIFIER

Once, the input data is ready in form of its embeddings the next step is to construct the classification model which will classify input data into different levels of Bloom's taxonomy. For our proposed classifier, LSTM has been chosen for its power in classifying sequences and text is a classical example of sequence. We have used a tagged dataset where the questions and CLOs are manually tagged as per their desired Bloom's taxonomy (cognitive level) categories. Therefore, our proposed classification model will classify a Question / CLO into the desired categories by LSTM network.

As, we have already discussed RNN in section II-D2 where we saw the problem of learning long-term dependencies in order to understand the context in the sequential data. This problem was called ''vanishing gradient'' [58]. There are cases in sequential data, where we need longer sequences in order to understand the context effectively. Let's consider predicting the last word in the text ''I grew up in France…I speak fluent **French**''. The recent information from the word ''speak'' and ''fluent'' indicates that the last word must be name of language. But, to understand or predict language name we need additional context upto the name of the country ''France''. Here, you can see the gap between the predicted word and the required context word is very large. Practically, RNN is not capable of solving these cases. This problem was identified by [67].

LSTM neural networks, are special kind of RNN networks which are capable of learning long-term sequence dependencies. These networks are specially designed to learn information for a long period of time. It has the ability to make decision regarding what information to keep and discard while processing input. Also, it has a gated mechanism to control the flow of the input sequences inside the LSTM cell. Before going into the detail of gating mechanism of LSTM, we need to understand that our proposed sequential neural network model initially processes input sequence in which the each word is represented as $w_1, w_2, \ldots . w_n$. Then, we have word embedding layer where input words are combined with 300-dimensional word vectors w_v and the output is given to the LSTM neural network as given in equation 6. The single LSTM network cell is shown in Figure 14. However, the network cell is step-by-step discussed further below.

$$output_d = w_v + w_n \tag{6}$$

The key element of the LSTM is its cell state. The cell state actually works like bridge for information flow. As shown in Figure 14, it is like a horizontal line running through the whole LSTM cell with some linear interactions where C_t is the new cell state and C_{t-1} is the old cell state. The output from the equation 6 will be flowing through this cell state.

Another important components of the LSTM network cells are its gates. These gates actually control the flow of information in different ways. As shown in Figure 14, the pink circle represents the pointwise multiplication operator and yellow box represents the sigmoid neural network layer. This layer

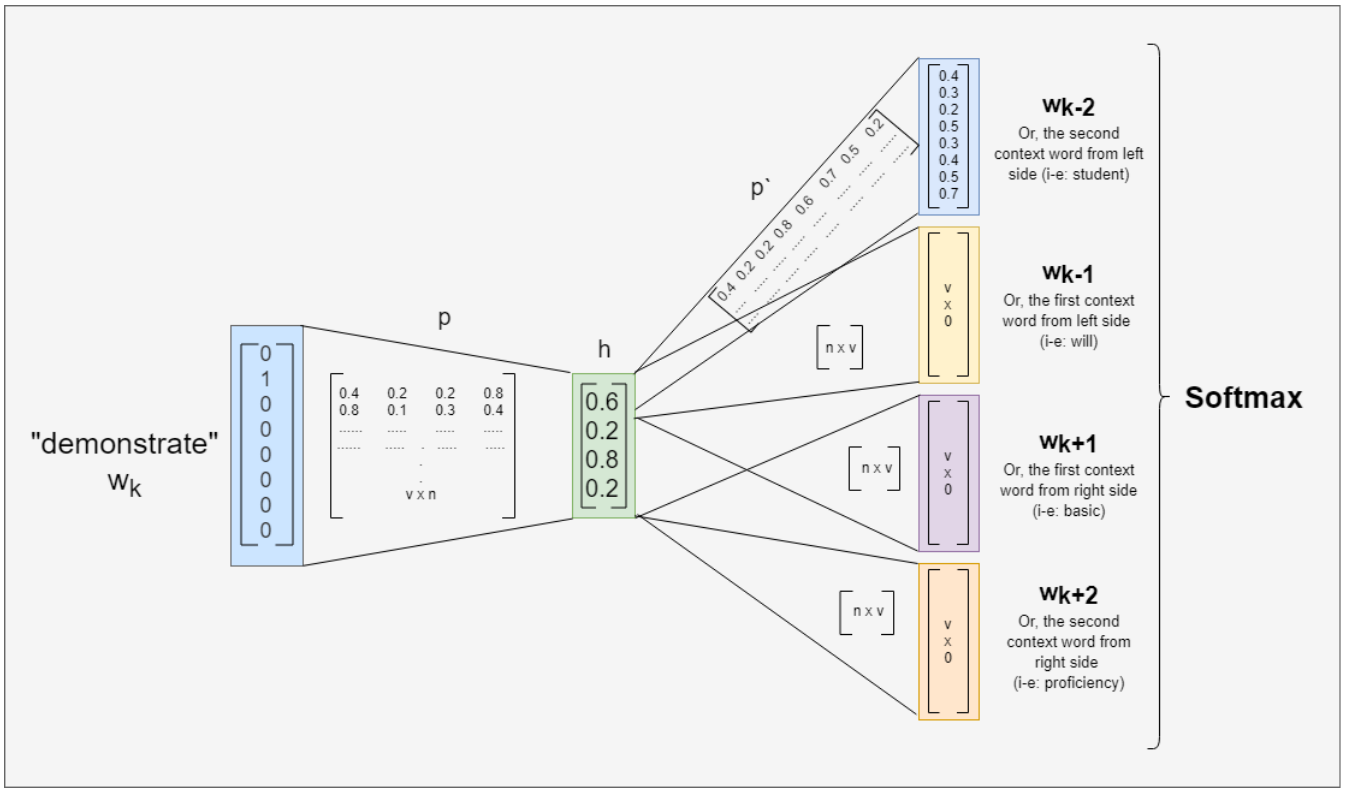

FIGURE 11. Sending w_k = "demonstrate" through the neural network to calculate softmax probabilities for context words ("student", "will", "basic", "proficiency").

```
word  [-2.3880e-01  5.6796e-02 -1.0488e-02  1.3781e-01 -8.2600e-02  3.1404e-02
 -3.9842e-01 -2.6896e-01  3.7160e-01 -1.9283e-01 -4.3051e-04  4.0367e-02 ……..]

following  [-0.23708    -0.36989    -0.22533    -0.036733   -0.16987
0.17781
  0.17734    -0.10996    -0.10268    -0.23101     0.0071532  -0.11403 ……..]

given [-3.3495e-01 -4.9739e-02 -1.9830e-01  9.9941e-02  6.7084e-02  1.6966e-01
 -1.2391e-01  1.7915e-02  3.5570e-01 -4.9377e-02  1.2104e-01 -7.1664e-02 ……..]

compare    [-3.1966e-02 -4.7873e-02 -1.7887e-01  2.3323e-01 -8.8079e-02
2.3159e-01
  9.5258e-02 -2.3224e-01  7.4979e-02  1.9977e-01  2.8826e-01 -2.0743e-01 ……..]

use   [-0.18815    0.22155   -0.29515    0.19578   -0.15808   -0.094363
  0.092856  -0.45765   -0.033427   0.13812   -0.21994   -0.1337 ……..]
```

FIGURE 12. Word embedding matrix representation for dataset 1 and dataset 2.

actually works on information control inside gates where sigmoid returns 0 if there is nothing needs to be done and 1 if there needs to be something done. Furthermore, LSTM network has three types of gates. The forget gate, the input gate and the output gate. All of these gates are further discussed below.

The first step after the information processing and entering into LSTM network is to decide what information needs to be excluded and what information needs to be considered for the network from the previous output state. This decision is made by forget gate by looking at previous output (h_{t-1}) and current input (x_t). As shown in Figure 14, the yellow box represents

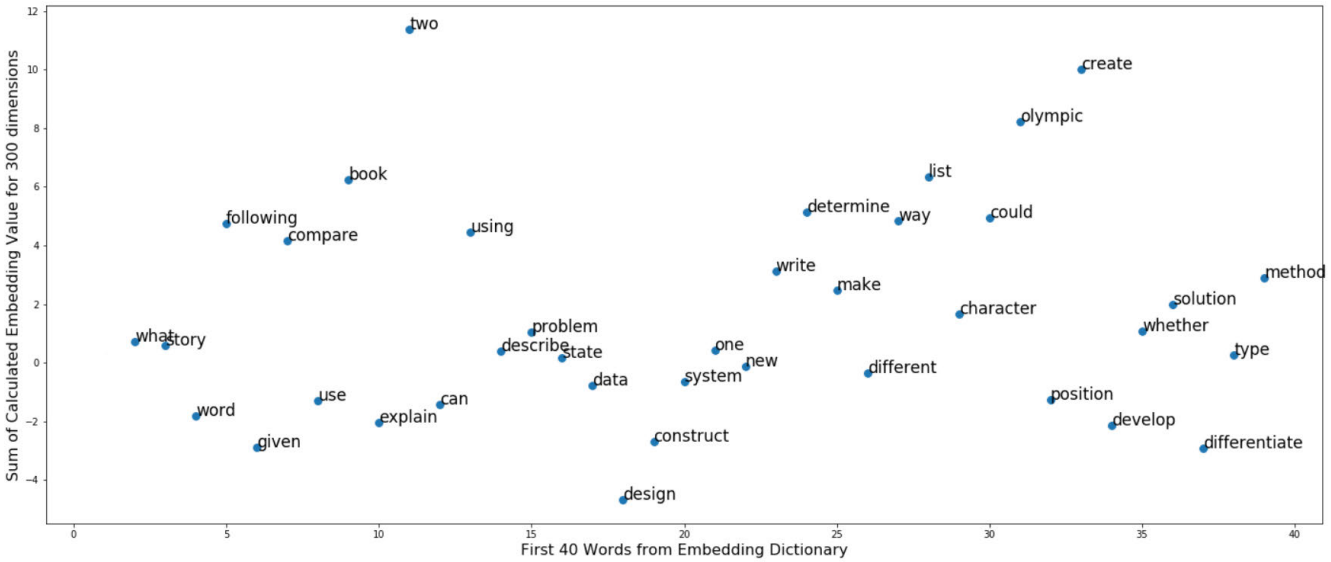

FIGURE 13. Neighbouring words representation for dataset 1 and dataset 2.

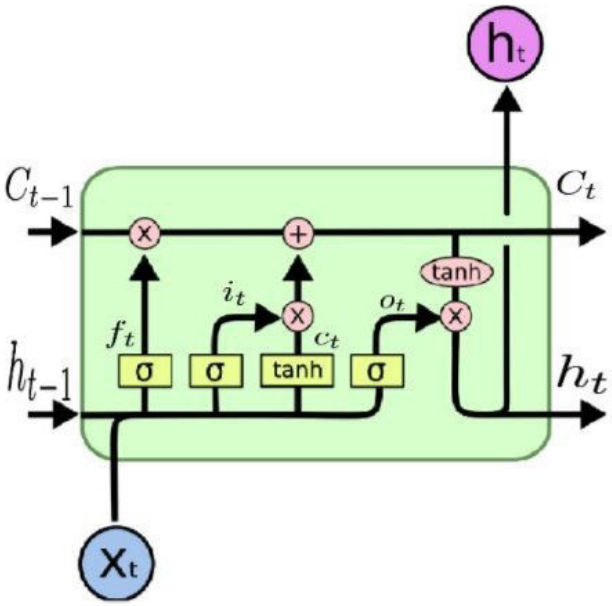

FIGURE 14. LSTM network cell with gating mechanism [60].

the sigmoid layer which results between 0 and 1 for each number in the cell state (C_{t-1}). 1 represents ''to consider the information'' and 0 represents ''to exclude the information'' from the network. The neural network equation for forget gate is given in equation 7. The forget remains empty initially as there is no previous output state.

$$f_t = \sigma\left(W_f \,.\, \left[h_{t-1}, x_t\right] + b_f\right) \tag{7}$$

where h_{t-1} is the output for previous state, x_t is the current input state. W_f, b_f are the weights, bias matrices for the forget gate, respectively.

Next, the LSTM network uses the input gate to decide what information needs to be added into the present cell state C_t from the current input. This input gate consists of two neural network layers namely, sigmoid and tanh layers as show in Figure 14. The sigmoid layer decides what values we'll update and the tanh layer creates a new vector comprises of new candidate values, $\widetilde{C}_t$ to be added into the present cell C_t. The sigmoid and tanh neural network layer equations for input gate are given in equation 8 and equation 9, respectively.

$$i_t = \sigma\left(W_i \,.\, \left[h_{t-1}, x_t\right] + b_i\right) \tag{8}$$

$$\widetilde{C}_t = tanh\left(W_C \,.\, \left[h_{t-1}, x_t\right] + b_C\right) \tag{9}$$

where h_{t-1} is the output for previous state, x_t is the current input state. W_i, b_i are the weights, bias matrices for the input gate sigmoid layer, respectively. Also, the W_C and b_C are the weights, bias matrices for the input gate tanh layer, respectively.

Once, the new candidate values vector is created in $\widetilde{C}_t$ it's time to update the old cell state C_{t-1} into the new cell state C_t, as shown in Figure 14. This is done by multiplying output from equation 7 (i-e: f_t) with output of previous state C_{t-1}. Next, the product of both input gate equations (i-e: equation 8 and 9) is also added into it. The output of the new cell state is given in equation 10.

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{10}$$

where f_t is the forget gate output, C_{t-1} is previous state output, i_t is the input gate output and $\widetilde{C}_t$ is the new candidate values' vector.

Finally, we need to decide what to generate for output. Here, LSTM uses its last gate ''the output gate''. This output gate decides to send specific information as output from the cell state C_t. As shown in Figure 14, the sigmoid layer in

output gate decides the part of information for the output, then the new cell state C_t is put into tanh layer to make the values between (-1 and 1). Lastly, the output of sigmoid layer is multiplied to just output the selected information. The sigmoid and tanh output layer are given in equation 11 and 12, respectively.

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{11}$$
$$h_t = o_t * tanh\left(C_t\right) \tag{12}$$

where h_{t-1} is the output for previous state, x_t is the current input state. W_o, b_o are the weights and bias matrices for the output gate sigmoid layer, respectively. Also, the C_t is the new cell state and h_t is the output after the output gate.

As the output from the hidden layer h_t is between $(-1$ and 1) and these values are not normalized. At the end, we need the probability distribution of total neurons defined at the output layer which are equal to the total class labels defined for classification. Therefore, the hidden state h_t output from LSTM layer is given to the dense connected output layer connected with softmax activation function which takes h_t and convert it into the normalized values over probability distribution of N possible outcomes. The maximum probability score for the redicted class label is selected as the desired input. The final output of our proposed LSTM neural network layer is given in equation 13.

$$Softmax\left(h_t\right) = \frac{\exp\left(h_t\right)}{\sum_{k=1}^{K}\exp\left(h_t\right)} \tag{13}$$

H. PROPOSED LSTM BASED SEQUENTIAL MODEL FOR CLASSIFICATION

The input to the LSTM network layer in our proposed model is the preprocessed Question / CLO statements combined with pre-trained embedding from ''Wiki Word Vectors'' of 300-dimensions. The output from LSTM network layer is probability distribution upto six Bloom's taxonomy labels (Remembering, Understanding, Application, Analysis, Evaluation and Creating). Figure 15 and 16 shows the summary of proposed LSTM model for CLOs and Question classification.

As, the size of dataset is small therefore we have used dropout rate of **0.2** at the LSTM layer in order to avoid model overfitting. The reason for the selection of **0.2** as the dropout value is where our proposed model performs best after several experiments. Moreover, in order to keep the efficient learning we have used **"Adam"** optimizer for CLOs classification model and **"RMS"** optimizer for Questions classification model because they usually work better for small datasets. Table 11 and Table 12 in the appendix show the process of selection for best optimizer, dropout value and batch size. Again, we have used a typical deep learning hit and trial process to fine tune these hyper parameter. We have applied **"Categorical Crossentropy"** loss function which works best for multicalss classification with balanced/imbalanced class distribution.

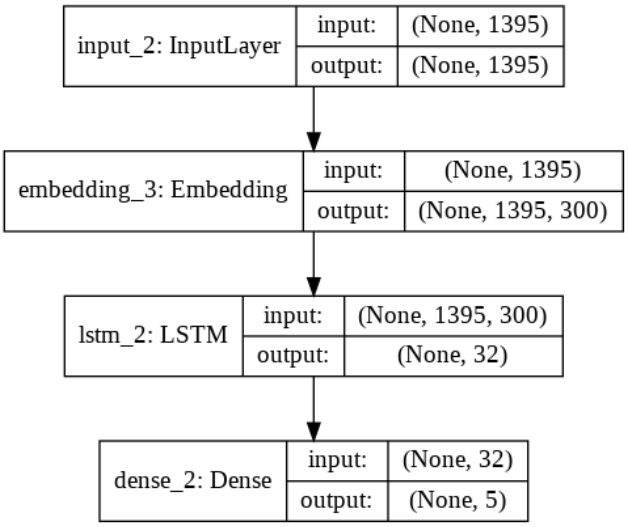

FIGURE 15. LSTM based CLOs classification model.

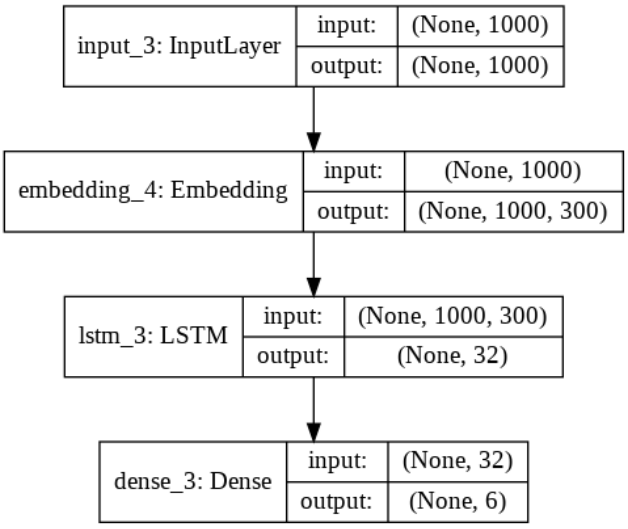

FIGURE 16. LSTM based question classification model.

I. EVALUATION OF PROPOSED CLASSIFICATION MODEL

The most common evaluation metrics used for evaluation of text classification models are Precision, Recall, F1-Score and Accuracy. The equations for calculating all these metrics are given in equations 14, 15, 16 and 17, respectively. All of these metrics are calculated by true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). These four numbers in combination creates a confusion matrix shown in Figure 17.

1) PRECISION

It is ratio of correctly predicted values for a specific class with respect to all predicted values in that class.

$$Precision = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{14}$$

	Predicted No	Predicted Yes
Actual No	TN	FP
Actual Yes	FN	TP

FIGURE 17. Confusion matrix.

2) RECALL

It is ratio of all predicted values for a specific class with respect to actual values in that class.

$$Recall = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{15}$$

3) F1-SCORE

It is a harmonic mean of precision and recall. It a balanced ratio of both precision and recall.

$$F1 - Score = \frac{2 \times (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{16}$$

4) ACCURACY

It is a ratio of correctly classified instances with respect to all values.

$$Accuracy = \frac{(\text{TP} + \text{TN})}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{17}$$

We have used average of class wise accuracy as a major evaluation metric to evaluate the proposed classification model for CLOs and Questions into Bloom's Taxonomy. However, we have also discussed other evaluation metrics including confusion matrix as well.

IV. EXPERIMENTAL RESULTS

This section shows the experimental settings and the detailed results obtained after the several experiments performed in our work. We explained different types of experiments performed for the evaluation of the proposed system.

A. USER DEFINED DATASET (DATASET 1)

The second component on which the proposed system depends is the CLOs tagged into Bloom's taxonomy (cognitive levels). Unfortunately, there is no such benchmark data-set available with ground truth values for this purpose. Therefore, we constructed a dataset of CLOs taken from the Sukkur IBA University. The department of quality enhancement cell (QEC) manually tagged 828 CLOs as per their Bloom's taxonomy level. These domain experts have been conducting different training for faculty members in understanding this tagging. The tagging was again verified from different faculty members from four departments (Computer Science, Electrical Engineering, Business Administration and Mathematics) using an online application where each faculty member logged in with its username and assign Bloom's level to the CLOs related to their courses. The CLOs were already available in their respective course outline / specification document, however, we had to spend time on its compilation in single table and manual tagging of these CLOs into their respective category. We used this dataset to create a baseline for upcoming researches in this field. The section III-C shows some of the major statistics for both datasets.

B. BENCHMARK QUESTIONS DATASET (DATASET 2)

The proposed system considerably depends on the variety of questions tagged into Bloom's taxonomy (cognitive levels). The questions must reflect maximum evaluation of student's learning based on Bloom's taxonomy levels. Therefore, we obtained such pre-built dataset from faculty members of Najran University, Saudi Arabia. In this dataset, the 600 questions are classified into six different levels of Bloom's taxonomy. We used this dataset as a baseline and evaluate our proposed system performance in terms of accuracy. The previous authors have reported classification accuracy of 84% in [31] and 89% in [36] using traditional machine learning approach.

C. EXPERIMENTAL SETTINGS

The table 2 depicts final experimental settings which we applied in order to get maximum accuracy. The accuracy measure is used here in order to evaluate the performance of both classification models. The model parameters mentioned in the table 2 are the final parameters for model training. For the best number of epochs selection process, we analyzed the overfitting and underfitting graphs as given in Figure 19. The Figure depicts that the model built for Dataset 1 has the highest accuracy at the epoch 8 and the model built for Dataset 2 has the highest accuracy at the epoch 25. Moreover, other technical parameters are finalized after several experiments as shown in table 9, 10, 11 and 12 in Appendix.

D. KEYWORDS BASED APPROACH RESULTS

This section explains the initial keyword based approach results, which we applied in order to set the baseline results. This is important to set baseline results because it will be used to evaluate the performance of proposed model. The reason to use this approach is because this is the original approach used in the literature and practical as well for Bloom's taxonomy classification. However, this approach suffers from one of the major problem explained in section I. To use this approach, initially we build the keywords/actions verbs dictionary representing six levels for the Bloom's taxonomy. We extracted major action verbs in each level of Bloom's taxonomy from different relevant sources. The list of these action verbs was already shown in Figure 2, previously.

Once, the dictionary is built we preprocessed and queried the CLOs and Questions statements from our datasets in order to search for action verbs/keywords from the dictionary.

TABLE 2. Classification models technical parameters.

Model	CLOs Classification	Question Classification
Dataset	Dataset 1	Dataset 2
Units (Input Layer)	1395	1000
Units (Embedding Layer)	1395	1000
Units (LSTM Layer)	32	32
Units (Output Layer)	5	6
Optimizer	Adam	RMS
Loss Function	Categorical Crossentropy	Categorical Crossentropy
Batch Size	16	16
Epochs	8	25
Activation (Output Layer)	Softmax	Softmax

The decision to assign Bloom's taxonomy level to CLO/Question was based on following steps.

- If the text contains single action verb with maximum frequency equal to 1 and it belongs to only one Bloom's level then the desired level is assigned to it.
- If the text contains single action verb with maximum frequency equal to 1 and it belongs to more than one Bloom's level then we randomized the Bloom's levels to assign the randomized level.
- If the text contains multiple action verbs with different frequencies then the verb with maximum frequency is used to get its Bloom's level and assigned it.
- If the text contains multiple action verbs with equal frequencies then again we randomized the Bloom's level to assign the randomized level.

The use of randomization is necessary here in order to make a decision. Because, in case of multiple levels for a single action verb we simply cannot decide the exact Bloom's level without understanding its context. Once the Bloom's level is assigned, it is validated using Bloom's level assigned by domain experts. This approach is further explained in algorithm 3.

Moreover, the results obtained using this approach for Dataset 1 and Dataset 2 are shown in table 3.

TABLE 3. Keyword based classification accuracy of both datasets.

Dataset	Accuracy	Total Instances
Dataset 1	40%	748
Dataset 2	54%	487

E. PROPOSED CLASSIFICATION MODEL EXPERIMENTAL RESULTS

This section explains different experimental results we observed while developing and evaluation of LSTM classification models for CLOs and questions using pre-trained wiki-word vectors. We applied two different approaches in order to evaluate accuracy metric for classification model namely Train/Test split and Cross Validation. The subsequent sections explain both of these approaches.

Algorithm 3 Keywords Based Approach Classification

INPUT: Raw CLO / Question Text
OUTPUT: Bloom's Taxonomy Level

```
keyword = k_1 ..... k_N
labelsList = lb1 ..... lb_N
preprocessedText ← pre − processing(INPUT)
keyword ← extractKeywords(preprocessedText)
N = length(keyword)
if N = 1 then
    BloomLevel ← searchBloomKeywordsDictionary(keyword)
    OUTPUT = BloomLevel
else
    for k ← 1 to k ← N do
        BloomLevel ← searchBloomKeywordsDictionary(k)
        labelsList.append(BloomLevel)
    end for
    frequencyCount ← getBloomLevelsFrequency(labelsList)
    if frequencyCount are equal then
        randomBloomLevel ← randomize(labelsList)
        OUTPUT = randomBloomLevel
    else
        mostFrequentLevel ← mostFrequentLevel(labelsList)
        OUTPUT ← mostFrequentBloomLevel
    end if
end if
```

1) TRAINING, TEST SET PARTITIONS

To start with conducting experiments, the manually tagged Dataset 1 (CLOs Dataset) was divided into different proportions of training and test sets. Initially, to create a start point the ratio of 50:50 was kept for training and test sets. The model obtained the accuracy of 65% on this first proportion. To further assess whether this distribution has any impact on model training, we increased the proportion gradually and found that our proposed model is also improving while learning from more data. The highest accuracy of 74% was obtained on the proportion of 75:25. After this proportion, the model started to decrease the accuracy. We set this distribution as break point for training, test proportions because further increasing the proportions results in no significant increase in the accuracy.

Furthermore, we performed several experiments for Dataset 2 (Questions Dataset) as well. To start working with that dataset, initially we started with 50:50 ratio for training and test set. The model obtained the accuracy of 72%. Also, to check impact of training and test set proportions on model training we gradually changed the proportions and observed that the model is improving. Finally we stopped at the proportion of 95:5 for the training and test set where model got highest accuracy of 87%. This accuracy is 3% more than the research study [31] where the authors had performed the same task and reported highest accuracy of 84%. However, Mohammed Manal *et al.* applied basic machine learning techniques to perform same task on this same dataset in [36] and the reported accuracy was 89%; which is the

TABLE 4. Classification accuracy of both datasets.

Dataset	Accuracy	Precision	Recall	F1-Score
Dataset 1	74%	73%	74%	73%
Dataset 2	87%	80%	89%	82%

maximum accuracy as compared to our proposed work and reported previously in [31]. But, this suffers from the two major problems.

1) **Generalization**: In general terms, this is a ability of the trained model that how it performs for unseen/new data. Usually, deep learning models need large amount of data to achieve maximum generalization. Although, in our case the amount of data was small but still there is only 2% difference in our accuracy and accuracy reported by [36]. This shows that even on small amount of data our proposed model is generalized well. Also, the generalization achieved by deep learning model is more effective than the one achieved by traditional machine learning models due to some of its benefits like automatic feature learning, hierarchical layer architectures, use of word embeddings, etc [68].
2) **Scalability**: Usually, we need large amount of data for the deep learning models to perform well. This is one of the main bottleneck for our proposed approach when we applied it on dataset 2 and got lower accuracy than the approach proposed in [31]. In future, we can improve our proposed approach with more data and we can easily beat the traditional machine learning models because these models are not very good when it comes to data scalability.

The accuracy, precision, recall and F1-score for both the datasets for our proposed approach are shown in Table 4. For dataset 1, we have shown the weighted-average of all the metrics because it is highly imbalanced. However, for dataset 2 we have shown macro-average of all the metrics due to its balanced nature. Furthermore, Figure 18 depicts relation of different training set proportions with test set accuracy.

Initially, the built model for CLOs classification using Dataset 1 with the proportion of 75:25 for training and test set was trained with 25 epochs. But, Figure 19 shows that there is no significant increase in accuracy after the 8 epochs. Therefore, we trained the CLOs classification model for 8 epochs. However, the model built for question classification using Dataset 2 with the proportion of 95:5 for training and test set was trained with 30 epochs at the start. But, Figure 19 depicts that the highest accuracy the model obtained was at the 20 epochs. Therefore, we trained the question classification model for 20 epochs only. Figure 19 explains the behaviour of model learning in terms of accuracy with different no. of epochs.

2) K-FOLD CROSS VALIDATIONS

In this research study we have tried to implement deep learning based classification models on small real world

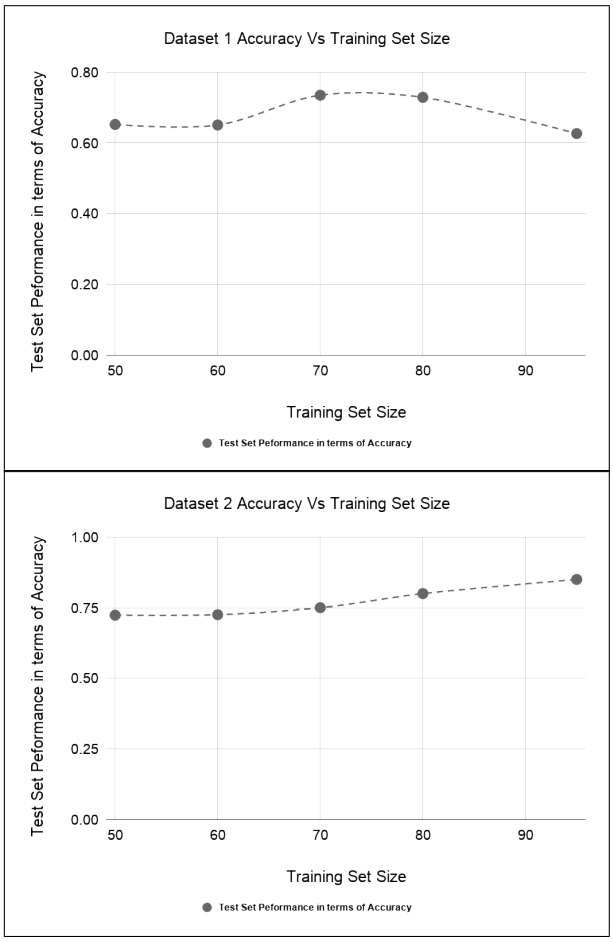

FIGURE 18. Performance of test set with different training set proportions.

TABLE 5. Classification accuracy of K = 10 fold cross validations.

Dataset	Accuracy	Total K Folds
Dataset 1	69%	10
Dataset 2	81%	10

datasets. Therefore, we have also evaluated its performance using k-fold cross validations in addition to train / test partitions. This technique randomly divides dataset into K distinct chunks where K-1 chunks are used to train the model and K chunk is kept as unseen in order to test the model performance using accuracy metric. We applied K = 10, fold cross validations in order to understand the model behaviour over both small datasets (i-e: Dataset 1 and Dataset 2). Table 5 depicts accuracy of both models where we took two highest accuracy values in 10 folds and reported its average.

F. COMPARISON WITH STATE-OF-THE-ART RESULTS FROM OTHER AUTHORS

We conducted a state-of-the-art analysis to compare the results of other existing techniques from other authors for classifying CLOs or exam questions items into Bloom's

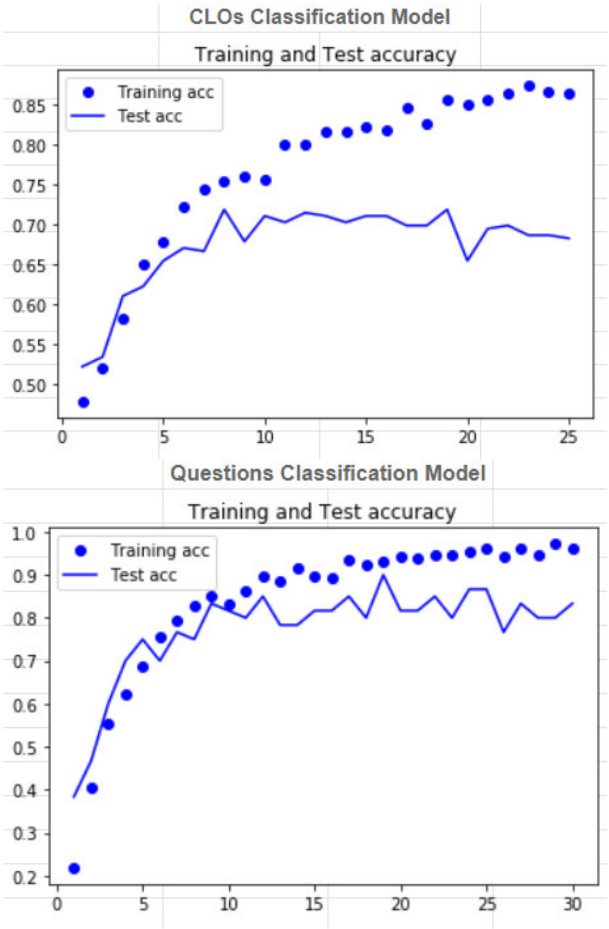

FIGURE 19. Model learning behaviour with different number of epochs.

taxonomy (cognitive domain). Table 6 shows the details of this comparison analysis. The main objective of all these studies was to classify assessment/questions items into Bloom's taxonomy (cognitive domain), which is actually the original objective of our research study as well.

TABLE 6. Comparison of existing techniques from other authors.

Reference	Year	Technique	Accuracy
[23]	2021	Machine learning approach	82%
[35]	2021	CNN + LSTM	80%
[36]	2020	KNN, Logistic Regression, SVM	83.70%
[17]	2017	Ensemble approach	82%
[1]	2012	Rule-based approach	45%
[31]	2011	SVM	65%
[32]	2010	SVM	76%
[21]	2008	Keyword-based approach	47%
[30]	2004	SVM	80%
[33]	2003	Artificial Neural Network	60%

G. OTHER EXPERIMENTAL RESULTS

As, stated earlier that due to small size of data we cannot learn efficient word vectors representations from data itself. Therefore, we used "Wiki-Word-Vectors" pre-trained embeddings in order to learn efficient neighbouring based context word representations. The other research studies in pre-trained embeddings domain also suggest the use of two famous pre-trained embeddings Glove.6B.300D and GoogleNews-vectors-negative300D as well. Therefore, in order to evaluate the performance of our proposed pre-trained embedding we performed experiments with same classification models and dataset distributions with Glove.6B.300D and GoogleNews-vectors-negative300D pre-trained embeddings. Table 7, shows the classification performance of the proposed pre-trained embedding with two other pre-trained embeddings. Furthermore, we know that for small amount of data most of the time traditional machine learning classifiers works very well. Therefore, we tried to compare the performance of our proposed classification model with state-of-the-art machine learning algorithms used for text classification. We used four traditional machine learning algorithms (SVM, Multinomial Naive Bayes, Logistic Regression and Random Forest) and two deep learning networks (Dense and SimpleRNN). The dataset was distributed in the same proportion as our proposed classification model. We used same pre-trained embedding used in our proposed classification model for Dense and Simple RNN networks. However, for traditional machine learning algorithms, we used the TF*IDF approach for feature representation. The table 8 depicts performance of all these other algorithms in terms of accuracy. The other models have perfomed very well but are still lower in accuracy as compared to our proposed model.

Figure 20 and 21 depicts the overall results of other experiments which we performed to evaluate the proposed model for CLOs and Question Classification into Bloom's Taxonomy.

TABLE 7. Comparison of different pre-trained word embeddings.

Word Embedding	Accuracy CLOs	Accuracy Questions
Glove.6B.300D	67%	77%
GoogleNews-vectors-negative300D	65%	60%
Proposed Pre-trained Embedding	**74%**	**87%**

TABLE 8. Comparison of different state-of-the-art algorithms.

Classifier	Accuracy CLOs	Accuracy Questions
SVM	65%	76%
Multinomial Naive Bayes	60%	70%
Logistic Regression	62%	77%
Random Forest	69%	79%
SimpleRNN	57%	13%
Dense Network	4%	43%
Proposed LSTM Model	**74%**	**87%**

H. CONFUSION MATRIX FOR PROPOSED LSTM MODEL

Figure 22 and Figure 23 show the confusion matrices of our proposed classification model for CLOs and Questions Classification. Figure 22 shows the confusion matrix for proposed LSTM model for CLOs classification. As shown here, out of 16 CLOs belonging to Remembering class, 14 were correctly classified. However, the only 2 instances were incorrectly classified into Understanding class. The Understanding

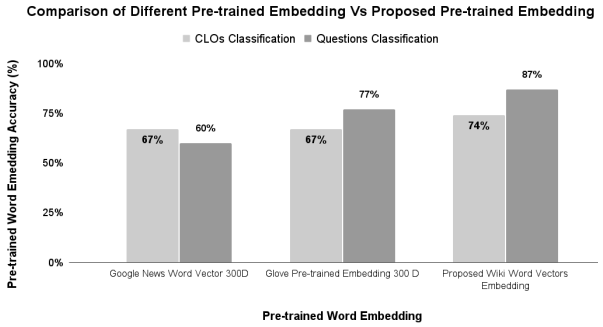

FIGURE 20. Comparison of different state-of-the-art Vs proposed (Pre-trained embedding).

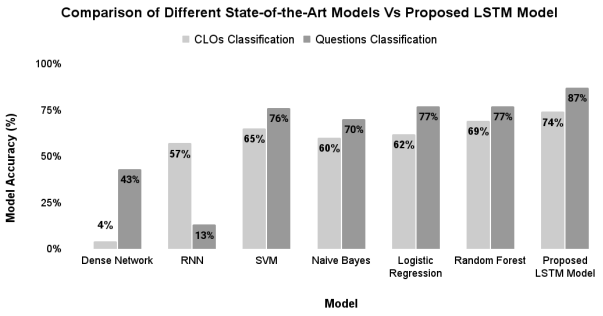

FIGURE 21. Comparison of different state-of-the-art models Vs proposed LSTM model.

and Application class contains major proportion of the test set with total 71 and 54, respectively. Out of these total instances, 61 and 46 were correctly classified, respectively. The remaining 26 instances out of 167 test set instances belongs to Analysis and Evaluation class. Here, the proposed model correctly classified the 19 and 1 CLOs into respective classes. We can see from the diagonal of the confusion matrix which actually represents maximum number of correctly classified class-wise instances even though three of total five classes contains very few instances.

However, Figure 23 shows the confusion matrix for proposed LSTM model for Questions Classification. As shown in the confusion matrix, there are overall 6 classes in which total 30 questions are divided. There are 4 questions correctly classified out of 5 in three of six classes namely, Remembering, Understanding and Application respectively. The 10 questions from Analysis and Evaluations classes are 100% correctly classified. However, the last Creating class has showed the lowest inter class performance where only 3 questions are correctly classified out of 5 questions.

V. DISCUSSION

In the experimental results, we have evaluated our proposed LSTM model for classification of CLOs and Questions over categories of Bloom's taxonomy. The experimental results showed the proposed LSTM classification model in combination with Wiki Word Vectors Pre-trained word embeddings gives the best results as compared to other pre-trained

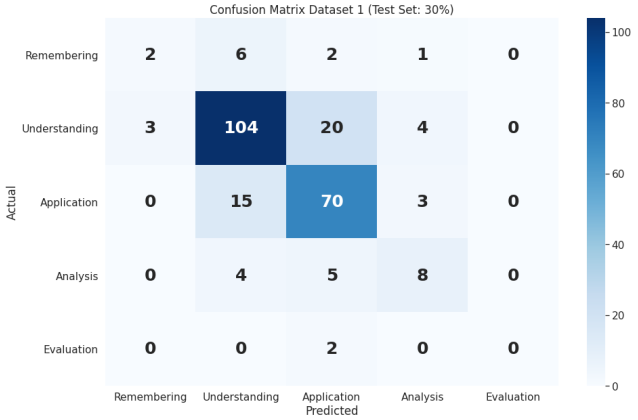

FIGURE 22. Confusion matrix CLOs classification model.

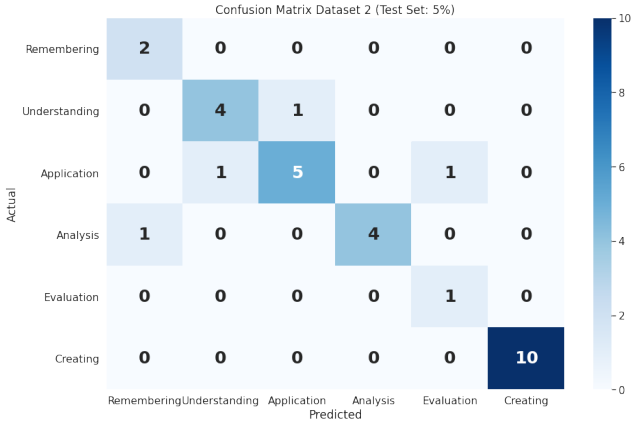

FIGURE 23. Confusion matrix questions classification model.

embeddings and state-of-the-art text classification models. The theoretical analysis of results is discussed in subsequent sections.

A. PROPOSED PRE-TRAINED WORD EMBEDDING

The use of pre-trained embeddings for word representation is an efficient way where the size of data is relatively small. In this study, we compared our proposed pre-trained word embedding namely "Wiki-Word Vectors" with other two pre-trained word embeddings namely, Glove.6B.300D and GoogleNews-vectors-negative300D. The experimental results showed that of these three embeddings, our proposed embedding outperformed. Conversely, the other two word embeddings exhibited lower results. The possible reason behind the best performance of "Wiki-Word Vectors" is that it considers neighbour words in order to understand context. This gives different representations for same word in different contexts. Moreover, it uses sub-word information by breaking down words into character level in order to compute word representations for out-of-vocabulary words as well. The details are explained in section III-E4.

The possible reason behind the lower performance of other two embeddings is due to its inability to handle unknown

or out-of-vocabulary words particularly in the academic domain. Although, these embeddings are continuous word representations in form of dense vectors but they do not take into account word morphology. Therefore, most of the out-of-vocabulary words are not computed from these embeddings and the representations become NULL. Overall, Table 7 and Figure 21 explains the performance comparison of three pre-trained embeddings.

B. PROPOSED LSTM CLASSIFICATION MODEL

Text classification using deep learning algorithms is a very active approach these days. Therefore, we employed deep learning based LSTM Network for our proposed classification model. In order to evaluate performance of our proposed model, we compared its performance with other state-of-the-art machine as well as deep learning classification models. The experimental results proved that our proposed LSTM Network achieved the excellent performance for the classification of CLOs and Questions into Bloom's Taxonomy as compared to other state-of-the-art classifiers like SVM, Multinomial Naive Bayes, Logistic Regression, Random Forest, SimpleRNN and Dense Network. The interesting point under discussion here is the traditional classification algorithms which we employed except SimpleRNN and Dense Network have shown reasonable accuracy but are still less than our proposed LSTM Model. However, the SimpleRNN and Dense Network has shown very poor results. The possible reasons may be that dense or simpleRNN network architectures are not efficient for learning longer text sequences. Our proposed model outperformed possibly because LSTM is efficient in order to consider long text sequences for context understanding. The actual problem in this classification task is the understanding of context because different categories of Bloom's taxonomy have same repeated action verbs. Only, the context understanding using longer sequences can solve this classification problem. In addition, the LSTM network has the ability to control different information using its gating mechanism as explained in section III-G. The table 8 and Figure 21 depicts the performance comparison of six state-of-the-art classification models.

C. CLASS WISE PERFORMANCE FOR PROPOSED SYSTEM

We developed two model variations from our proposed classification model, one with 5 classes (i-e: CLOs Classification Model) and another one with 6 classes (Questions Classification Model) based on the manually tagged datasets (Dataset 1 and Dataset 2), respectively. As shown in Figure 22, the results showed that the CLOs classification model performed very well for first 4 classes (i-e Remembering - Analysis) by giving precision of 85%+. But, the last class (Evaluation) gives the precision of 33% which overall reduces the average performance of the model. The reasons for these poor results for last class is the unavailability of tagged CLOs for this specific level. Because, this class only contained total 10 instances, out of which 7 were used for training set and 3 were used for test set according to 75:25 training / test ratio. The precision for this class and overall performance of the model can be increased with the availability of more tagged data for this specific level. Also, another reason for low performance results is the imbalanced data in all of 5 classes. Hence, fixing these issues might increase the overall average performance of the proposed model.

VI. LIMITATIONS

Although, the proposed model demonstrated reasonable performance in classifying CLOs and question items into Bloom's taxonomy but still it suffers from various limitations. For example, Figure 23, shows that the CLOs classification model performed exceptionally well for 2 classes (i-e Analysis and Evaluation) by giving precision of 100%. However, for the first three classes (i-e: Remembering - Application) the model gives overall precision of 80% which is also quite satisfactory. But, the last class (Evaluation) gives the precision of 60% which overall reduces the average performance of the model. The model is trained on CLOs and assessment items in English language only. Therefore it will be difficult for the model to predict the Bloom's taxonomy level for non-english statements. The proposed model is trained on CLOs and assessment items from different subjects including computer science, electrical engineering, social sciences. However, for the CLOs and assessment items for subjects from medical and law domain, the model can perform low.

VII. CONCLUSION

The categorization of CLOs and Questions into Bloom's taxonomy is a purely domain expert task because it involves thorough understanding of assigning specific Bloom's taxonomy level in order to maximize the student's learning. The manual task is actually time consuming, laborious and often leads to mistakes due to human biasness. In this research, an automatic classification system is proposed for the classification of CLOs and Questions into Bloom's taxonomy using domain understanding. The categories used for classification were Remembering, Understanding, Application, Analysis, Evaluation and Creating. Our proposed model initially understood the domain for each different level using a manually tagged datasets of CLOs and Questions into Bloom's taxonomy levels. The model adapted the domain understanding using skip-gram pre-trained embedding namely, ''Wiki-Word Vectors''. This takes into account the context of neighbor words. Once, the model adapted enough domain understanding then it started to classify the CLOs and Questions into specific category using single layer LSTM model. The performance of the model was evaluated by accuracy metric using train/test split and 10-fold cross validations. However, we also evaluated the model performance by comparing it with two other pre-trained word embeddings and six state-of-the-art classification algorithms from the literature. During all these comparisons, our proposed domain based word embedding and LSTM model outperformed. We obtained very encouraging results for CLOs Classification (74%) and Question Classification (87%) as the dataset was relatively small upto

just few hundreds instances. We also supported our model performance with an existing study whereas our proposed model achieved 3% increase in accuracy as compared to that study for the same task.

VIII. FUTURE WORK

This work of automatic classification of learning outcomes and questions into Bloom's taxonomy can be further extended by developing specific domain based word embeddings by collecting large amount of CLOs and questions. We can process that amount and build specific skip-gram based word embedding. Also, various other neural network architectures like GRU, Deep Memory Networks or Ensemble Deep Learning models can be evaluated rather than LSTM to evaluate the performance of the proposed system. As, this work is based on supervised classification therefore another work that can be done is to develop a standard tagged dataset of CLOs and Questions with balanced classes and thousand of instances. This will make the learning of deep learning classifier more efficient.

Another, natural language processing based approach can be used in which we can work on the role of meta data (i-e: length of the text, etc) in classification of those keywords which are overlapping. Also, another extension may be the use of voting classifier because our dataset is tagged in three different ways. 1) Human-labelled, 2) Keywords-based-labelled and 3) machine-learning-model-labelled. So, we can create a voting classifier on top of all these classifications. However, this classifier is only for cognitive category but there are two other categories of Bloom's taxonomy (i-e: Affective and Psychomotor). Same architecture can be used for these two categories by using same tagged dataset approach.

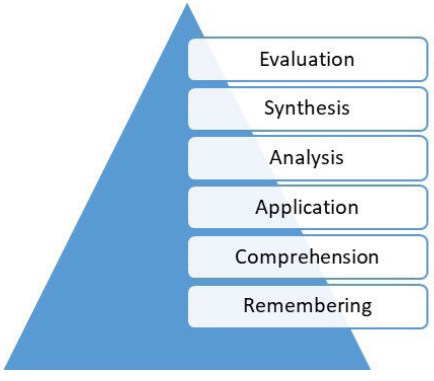

FIGURE 24. Bloom's taxonomy (Cognitive domain) hierarchy levels.

APPENDIX A
INTERVIEW QUESTIONS

1) Do you consider categorization of CLOs and examination questions into Bloom's taxonomy (Cognitive Levels) is a good way to assess student's learning outcomes in a specific course?
2) Do you think the teachers map the CLOs and examination questions into Bloom's taxonomy correctly?
3) Do you provide training to the desired faculty members for understanding of the domain while mapping? (Asked to accrediation bodies coordinator and HoDs)? If yes? Does the training is sufficient for new comer in this domain?
4) Were you provided training of mapping CLOs and examination questions into Bloom's taxonomy as soon as after your joining?
   If yes? Was that training sufficient for understanding the domain?
5) Do you only use keyword-based approach for assigning Bloom's level?

TABLE 9. Different configurations for selecting max length & unique words (CLOs classification model).

Dataset	Classification Model	Unique Words	Max Len	Train Acc:	Test Acc:
Dataset 1	ClOs Classification	1395	500	**87%**	**75%**
			1000	72%	67%
		1000	500	77%	67%
			1000	71%	69%
		500	500	72%	65%
			1000	74%	68%

TABLE 10. Different configurations for selecting max length & unique words (Questions classification model).

Dataset	Classification Model	Unique Words	Max Len	Train Acc:	Test Acc:
Dataset 2	Questions Classification	1000	500	**95%**	**87%**
			1000	94%	76%
		500	500	91%	73%
			1000	91%	66%

TABLE 11. Different configurations for selecting batch size, optimizer and regularizer (CLOs classification model).

	Optimizers	Batch Size	Regularizers	Train Acc:	Test Acc:
ClOs Classification	Adam	16	Dropout (0.2)	**87%**	**75%**
			Dropout (0.5)	64%	59%
			Dropout (0.9)	5%	7%
		8	Dropout (0.2)	81%	67%
			Dropout (0.5)	69%	62%
			Dropout (0.9)	5%	7%
	SGD	16	Dropout (0.2)	51%	49%
			Dropout (0.5)	51%	49%
			Dropout (0.9)	6%	4%
		8	Dropout (0.2)	51%	49%
			Dropout (0.5)	49%	54%
			Dropout (0.9)	63%	40%
	RMS	16	Dropout (0.2)	73%	68%
			Dropout (0.5)	63%	63%
			Dropout (0.9)	6%	4%
		8	Dropout (0.2)	76%	69%
			Dropout (0.5)	65%	67%
			Dropout (0.9)	63%	4%

TABLE 12. Different configurations for selecting batch size, optimizer and regularizer (Questions classification model).

	Optimizers	Batch Size	Regularizers	Train Acc:	Test Acc:
Questions Classification	RMS	16	Dropout (0.2)	**95%**	**87%**
			Dropout (0.5)	76%	73%
			Dropout (0.9)	17%	20%
		8	Dropout (0.2)	96%	73%
			Dropout (0.5)	83%	73%
			Dropout (0.9)	39%	46%
	Adam	16	Dropout (0.2)	94%	76%
			Dropout (0.5)	78%	73%
			Dropout (0.9)	16%	23%
		8	Dropout (0.2)	96%	80%
			Dropout (0.5)	85%	69%
			Dropout (0.9)	17%	3%
	SGD	16	Dropout (0.2)	48%	43%
			Dropout (0.5)	35%	26%
			Dropout (0.9)	17%	10%
		8	Dropout (0.2)	61%	53%
			Dropout (0.5)	45%	36%
			Dropout (0.9)	17%	3%

If yes? Then how you resolve the issue of overlapping keywords in different levels?
If no? Then what is another alternate approach for performing this activity?

6) What do you mainly check, once the subject specialist brings mapped CLOs and examination question to you for assessment? (Asked to coordinators of accrediation bodies)
7) Do you think, we can automate this categorization process using recent technologies?

REFERENCES

[1] N. Omar, S. S. Haris, R. Hassan, H. Arshad, M. Rahmat, N. F. A. Zainal, and R. Zulkifli, "Automated analysis of exam questions according to Bloom's taxonomy," *Procedia Social Behav. Sci.*, vol. 59, pp. 297–303, Oct. 2012.

[2] W.-C. Chang and M.-S. Chung, "Automatic applying Bloom's taxonomy to classify and analysis the cognition level of english question items," in *Proc. Joint Conf. Pervas. Comput. (JCPC)*, Dec. 2009, pp. 727–734.

[3] A. OSMAN and A. A. Yahya, "Classifications of exam questions using natural language syntatic features: A case study based on Bloom's taxonomy," in *Proc. 3rd Int. Arab Conf. Qual. Assurance Higher Educ.*, 2016, pp. 1–8.

[4] S. U. Monrad, N. L. B. Zaidi, K. L. Grob, J. B. Kurtz, A. W. Tai, M. Hortsch, L. D. Gruppen, and S. A. Santen, "What faculty write versus what students see? Perspectives on multiple-choice questions using Bloom's taxonomy," *Med. Teacher*, vol. 43, pp. 1–12, Jan. 2021.

[5] B. S. Bloom, "Taxonomy of educational objectives: The classification of educational goals," *Cognit. Domain*, vol. 51, no. 4, pp. 441–453, Dec. 1981.

[6] D. R. Krathwohl and L. W. Anderson, "Merlin C. Wittrock and the revision of Bloom's taxonomy," *Educ. Psychologist*, vol. 45, no. 1, pp. 64–65, Jan. 2010.

[7] S.-Y. Chyung and D. Stepich, "Applying the 'congruence' principle of Bloom's taxonomy to designing online instruction," *Quart. Rev. Distance Educ.*, vol. 4, no. 3, pp. 317–330, 2003.

[8] 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, p. 447, 2015.

[9] S. S. Haris and N. Omar, "A rule-based approach in Bloom's taxonomy question classification through natural language processing," in *Proc. 7th Int. Conf. Comput. Converg. Technol. (ICCCT)*, Dec. 2012, pp. 410–414.

[10] K. Jayakodi, M. Bandara, and I. Perera, "An automatic classifier for exam questions in engineering: A process for Bloom's taxonomy," in *Proc. IEEE Int. Conf. Teaching, Assessment, Learn. Eng. (TALE)*, Dec. 2015, pp. 195–202.

[11] 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. 4, pp. 142–149, 2016.

[12] N. N. Khairuddin and K. Hashim, "Application of Bloom's taxonomy in software engineering assessments," in *Proc. 8th WSEAS Int. Conf. Appl. Comput. Sci.*, 2008, pp. 66–69.

[13] K. Osadi, M. G. N. A. S. Fernando, and W. Welgama, "Ensemble classifier based approach for classification of examination questions into Bloom's taxonomy cognitive levels," *Int. J. Comput. Appl.*, vol. 975, p. 8887, Mar. 2017.

[14] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, Apr. 2019.

[15] S. K. Dwivedi and C. Arya, "Automatic text classification in information retrieval: A survey," in *Proc. 2nd Int. Conf. Inf. Commun. Technol. Competitive Strategies (ICTCS)*, 2016, p. 131.

[16] S. Das, S. K. Das Mandal, and A. Basu, "Classification of action verbs of Bloom's taxonomy cognitive domain: An empirical study," *J. Educ.*, vol. 201, Apr. 2021, Art. no. 002205742110021.

[17] Y. Bengio and J.-S. Senecal, "Adaptive importance sampling to accelerate training of a neural probabilistic language model," *IEEE Trans. Neural Netw.*, vol. 19, no. 4, pp. 713–722, Apr. 2008.

[18] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.

[19] J. Zhang, C. Wong, N. Giacaman, and A. Luxton-Reilly, "Automated classification of computing education questions using Bloom's taxonomy," in *Proc. Australas. Comput. Educ. Conf.*, Feb. 2021, pp. 58–65.

[20] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proc. 26th Int. Conf. World Wide Web Companion (WWW) Companion*, 2017, pp. 759–760.

[21] A. J. Swart and M. Daneti, "Analyzing learning outcomes for electronic fundamentals using Bloom's taxonomy," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, Apr. 2019, pp. 39–44.

[22] A. N. Rahmatih, D. Indraswati, G. Gunawan, A. Widodo, M. A. Maulyda, and M. Erfan, "An analysis of questioning skill in elementary school pre-service teachers based on Bloom's taxonomy," *J. Phys., Conf. Ser.*, vol. 1779, no. 1, Feb. 2021, Art. no. 012073.

[23] K. Atiullah, S. W. Fitriati, and D. Rukmini, "Using revised Bloom's taxonomy to evaluate higher order thinking skills (hots) in reading comprehension questions of english textbook for year X of high school," *English Educ. J.*, vol. 9, no. 4, pp. 428–436, 2019.

[24] B. D. Wijanarko, Y. Heryadi, H. Toba, and W. Budiharto, "Question generation model based on key-phrase, context-free grammar, and Bloom's taxonomy," *Educ. Inf. Technol.*, vol. 26, no. 2, pp. 2207–2223, Mar. 2021.

[25] D. Hoogeveen, L. Wang, T. Baldwin, and K. M. Verspoor, "Web forum retrieval and text analytics: A survey," *Found. Trends Inf. Retr.*, vol. 12, no. 1, pp. 1–163, 2018.

[26] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008, ch. 20, pp. 405–416.

[27] C. Buckley, "Implementation of the smart information retrieval system," Cornell Univ., New York, NY, USA, Tech. Rep. TR85-686, 1985. [Online]. Available: http://techreports.library.cornell.edu:8081/Dienst/UI/1.0/Display/cul.cs/TR85-686

[28] C. O'Riordan and H. Sorensen, "Information filtering and retrieval: An overview," in *Proc. 16th Annu. Int. Conf.*, Atlanta, GA, USA, Oct. 1997, pp. 28–31.

[29] J. Harrison, O. Dikken, and D. Peer, "Question classification according to Bloom's revised taxonomy," M.S. thesis, Bachelor End Project Rep., Jun. 2017.

[30] M. J. W. van Hoeij, J. C. M. Haarhuis, R. F. A. Wierstra, and P. van Beukelen, "Developing a classification tool based on Bloom's taxonomy to assess the cognitive level of short essay questions," *J. Vet. Med. Educ.*, vol. 31, no. 3, pp. 261–267, Sep. 2004.

[31] A. A. Yahya and A. Osman, "Automatic classification of questions into Bloom's cognitive levels using support vector machines," Naif Arab Univ. Secur. Sci., Riyadh, Saudi Arabia, Tech. Rep. 2914, 2011.

[32] N. Yusof and C. J. Hui, "Determination of Bloom's cognitive level of question items using artificial neural network," in *Proc. 10th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2010, pp. 866–870.

[33] D. Zhang and W. S. Lee, "Question classification using support vector machines," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Informaion Retr. (SIGIR)*, 2003, pp. 26–32.

[34] C. Servin, C. Tang, M. Geissler, M. Stange, and C. Tucker, "Enhanced verbs for Bloom's taxonomy with focus on computing and technical areas," in *Proc. 52nd ACM Tech. Symp. Comput. Sci. Educ.*, Mar. 2021, p. 1270.

[35] 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," 2021, *arXiv:2104.08819*. [Online]. Available: http://arxiv.org/abs/2104.08819

[36] 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, Mar. 2020, Art. no. e0230442.

[37] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc.*, 2010, pp. 1–24.

[38] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[39] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1631–1642.

[40] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12 pp. 2493–2537, Aug. 2011.
[41] A. M. Dai, C. Olah, and Q. V. Le, "Document embedding with paragraph vectors," 2015, *arXiv:1507.07998*. [Online]. Available: http://arxiv.org/abs/1507.07998
[42] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, "Sentiment analysis is a big suitcase," *IEEE Intell. Syst.*, vol. 32, no. 6, pp. 74–80, Nov./Dec. 2017.
[43] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Intell. Res.*, vol. 37, pp. 141–188, Feb. 2010.
[44] J. Weston, S. Bengio, and N. Usunier, "WSABIE: Scaling up to large vocabulary image annotation," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1–7.
[45] I. Sindhu, S. Muhammad Daudpota, K. Badar, M. Bakhtyar, J. Baber, and M. Nurunnabi, "Aspect-based opinion mining on student's feedback for faculty teaching performance evaluation," *IEEE Access*, vol. 7, pp. 108729–108741, 2019.
[46] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[47] H. Zhu, I. C. Paschalidis, and A. Tahmasebi, "Clinical concept extraction with contextual word embedding," 2018, *arXiv:1810.10566*. [Online]. Available: http://arxiv.org/abs/1810.10566
[48] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805
[49] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.
[50] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.
[51] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 5528–5531.
[52] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 1017–1024.
[53] S. Liu, N. Yang, M. Li, and M. Zhou, "A recursive recurrent neural network for statistical machine translation," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, 2014, pp. 1491–1500.
[54] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks," Johns Hopkins Univ., USA, Tech. Rep. 86, 2013.
[55] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.
[56] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," 2014, *arXiv:1402.1128*. [Online]. Available: http://arxiv.org/abs/1402.1128
[57] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3128–3137.
[58] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, 1998.
[59] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Proc. 13th Annu. Conf. Int. Speech Commun. Assoc.*, 2012.
[60] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[61] F. A. Gers, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Netw., (ICANN)*, 1999, pp. 850–855.
[62] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555
[63] Y. Liu, C. Sun, L. Lin, and X. Wang, "Learning natural language inference using bidirectional LSTM model and inner-attention," 2016, *arXiv:1605.09090*. [Online]. Available: http://arxiv.org/abs/1605.09090
[64] G. N. R. Prasad, "Evaluating student performance based on Bloom's taxonomy levels," *J. Phys., Conf. Ser.*, vol. 1797, no. 1, Feb. 2021, Art. no. 012063.
[65] I. J. Unanue, E. Z. Borzeshi, and M. Piccardi, "Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition," *J. Biomed. Inform.*, vol. 76, pp. 102–109, Dec. 2017.
[66] R. Collobert and J. Weston, "A unified architecture for natural language processing," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 160–167, doi: 10.1145/1390156.1390177.
[67] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
[68] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," 2017, *arXiv:1706.08947*. [Online]. Available: http://arxiv.org/abs/1706.08947

SARANG SHAIKH received the master's degree in computer science from Sukkur IBA University, Pakistan, in 2020. He is currently pursuing the Ph.D. degree with the Department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU), Norway. Prior to joining NTNU University, he was employed as a Visiting Lecturer with Sukkur IBA University. He is the author of several articles published in international journals. His research interests include applied research in the field of artificial intelligence, NLP, machine learning, deep learning, and learning technologies. He served as a Reviewer for IEEE Access.

SHER MUHAMMAD DAUDPOTTA received the master's and Ph.D. degrees from Asian Institute of Technology, Thailand, in 2008 and 2012, respectively. He is currently serving as a Professor of computer science for Sukkur IBA University, Pakistan. Alongside his computer science contribution, he is also a Quality Assurance Expert in higher education. He has reviewed more than 50 universities in Pakistan for quality assurance on behalf of the Higher Education Commission in the role of educational quality reviewer. He is the author of more than 35 peer-reviewed journal and conference publications. His research interests include deep learning, natural language processing, and video and signal processing.

ALI SHARIQ IMRAN (Member, IEEE) received the master's degree in software engineering and computing from the National University of Sciences and Technology (NUST), Pakistan, in 2008, and the Ph.D. degree in computer science from the University of Oslo (UiO), Norway, in 2013. He is currently associated with the Department of Computer Science, Norwegian University of Science and Technology (NTNU), Norway, as an Associate Professor. He specializes in applied research with a focus on deep learning technology and its application to signal processing, natural language processing, and the semantic web. He has over 65 peer-reviewed journals and conference publications to his name. He is a member of Norwegian Colour and Visual Computing Laboratory (Colourlab), and an IEEE Member, Norway Section. He served as a Reviewer for many reputed journals over the years, including IEEE Access, as an Associate Editor.