

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

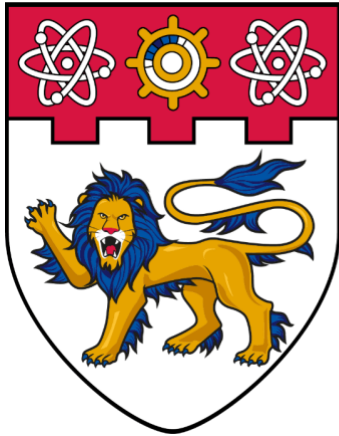
SINGAPORE

**Automated Scoring for Short Questions with Deep
Learning**

Submitted by
Michelle Vanessa
U1620071L

Project Supervisor
Assoc Prof. Hui Siu Cheung

School of Computer Science and Engineering
AY2019/2020



**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

SCE19-0305

**Automated Scoring for Short Questions with Deep
Learning**

Submitted in Partial Fulfilment of the Requirement for the Degree of Bachelor
of Engineering (Computer Science)

Submitted by
Michelle Vanessa
U1620071L

Project Supervisor
Assoc Prof. Hui Siu Cheung

School of Computer Science and Engineering
AY2019/2020

Abstract

This report proposes an improved approach of short-answered questions scoring with deep learning, namely Siamese Bidirectional LSTM model with feature engineering. The project will conduct several experiments by applying various methods on the model and dataset. The performance of each model in the experiments is evaluated and compared to obtain the optimal result. The scope of this experiment is limited to short-answered Data Structures questions, and the proposed improved approach provides a better performance in grading answers, measured in Pearson correlation coefficient, Rooted Mean Square Error, and Mean Absolute Error.

Acknowledgements

The author would like to express her gratitude to various individuals for their continuous encouragement. This project would not have been possible without their support.

The author would like to extend her deepest gratitude to Assoc Prof Hui Siu Cheung, the author's final year project supervisor, for his guidance, encouragement, and constructive feedbacks throughout the year.

Furthermore, the author would also like to express special thanks to her parents who have supported her throughout her journey in Nanyang Technological University Singapore. Without them, this project would not have been as successful.

Lastly, the author wishes to thank all her friends who have supported her during her time in university and have been there for her when she needed encouragement. Without their faith in her, the author would not have been able to complete this project as smoothly.

Table of Contents

<i>Abstract.....</i>	<i>2</i>
<i>Acknowledgements.....</i>	<i>3</i>
<i>Table of Contents</i>	<i>4</i>
<i>List of Figures</i>	<i>7</i>
<i>List of Equations</i>	<i>9</i>
<i>List of Tables</i>	<i>10</i>
<i>Chapter 1: Introduction.....</i>	<i>11</i>
1.1 Background.....	11
1.2 Motivation.....	11
1.3 Objectives.....	11
1.4 Scope.....	12
<i>Chapter 2: Literature Review</i>	<i>13</i>
2.1 Past Works	13
2.2 Multilayer Perceptron.....	13
2.3 Batch Normalization	14
2.4 Dropout	14
2.5 Question Demoting.....	14
2.6 Recurrent Neural Network (RNN)	14
<i>Chapter 3: Existing Approach.....</i>	<i>16</i>
3.1 Architecture	16
3.1.1 Embedding Layer	16
3.1.2 Vertical Layers.....	17
3.1.3 Concatenation Layer	19
3.1.4 Perceptron Layers.....	19
3.1.5 Output Layer	20
3.2 Bidirectional LSTM Unit.....	20
3.2.1 Forget gate.....	20
3.2.2 Input gate.....	20

3.2.3	Output gate	21
Chapter 4:	<i>Experimental Evaluation</i>	23
4.1	Dataset	23
4.2	Data Pre-Processing	23
4.3	Evaluation Metrics	23
4.3.1	Pearson Correlation Coefficient	23
4.3.2	Rooted Mean Square Error	24
4.3.3	Mean Absolute Error	25
4.4	Data Augmentation	25
4.5	Evaluation	26
4.5.1	Implementation	26
4.5.2	Model Selection	27
4.5.3	Training	27
4.6	Performance of Existing Model	27
4.7	Improving Model Performance	29
4.7.1	Batch Normalization	29
4.7.2	Regularization	30
4.7.3	Question Demoting	31
4.7.4	Classification Model	31
4.8	Experiment Results	34
4.8.1	Recurrent Batch Normalization Results	34
4.8.2	Regularization Results	34
4.8.3	Question Demoting	37
4.8.4	Classification Results	38
Chapter 5:	<i>System Implementation</i>	40
5.1	Software Architecture	40
5.1.1	Design Pattern	40
5.1.2	Frameworks	40
5.1.3	Database	41
5.2	System Interface	41
5.2.1	General	42
5.2.2	Admin	43
5.2.3	Student	48
Chapter 6:	<i>Conclusion</i>	51

<i>Reference</i>	52
<i>Appendix A: Experiment Results (Regression Model)</i>	56

List of Figures

Figure 1 Network architecture illustration	16
Figure 2 Multilayer perceptron (tokenization).....	17
Figure 3 Multilayer perceptron (feature engineering)	18
Figure 4 Perceptron layers	19
Figure 5 LSTM Unit	21
Figure 6 LSTM network	21
Figure 7 Bidirectional LSTM network	22
Figure 9 Pearson correlation coefficient	24
Figure 10 Training error of original model.....	28
Figure 11 Validation error of original model.....	28
Figure 12 Dropout on neural network with 2 hidden layers	31
Figure 13 Classifier terms	33
Figure 14 Training loss of optimal model.....	38
Figure 15 Validation loss of optimal model	38
Figure 16 Django architecture pattern	40
Figure 17 Django and React with Django REST Framework	41
Figure 18 Sign up page	42
Figure 19 Log in page	43
Figure 20 Admin questions list	43
Figure 21 Admin question details	44
Figure 22 Add questions	44
Figure 23 Admin posts list	45
Figure 24 Admin post details	45
Figure 25 Admin answers list	46
Figure 26 Add answers	46
Figure 27 Model details	47
Figure 28 Students list	47
Figure 29 Student details	48
Figure 30 Student questions list	48
Figure 31 Student answer question page	49
Figure 32 Student posts list.....	49

Figure 33 Student account details	50
Figure 34 Student edit account	50

List of Equations

Equation 1 Distance between two vectors	19
Equation 2 Forget gate	20
Equation 3 Sigmoid function	20
Equation 4 Input gate sigmoid layer	20
Equation 5 Input gate tanh layer	20
Equation 6 Tanh function.....	20
Equation 7 Output gate	21
Equation 8 Cell state	21
Equation 9 LSTM output	21
Equation 10 Min-max normalization	23
Equation 11 Pearson correlation coefficient	23
Equation 12 Covariance	24
Equation 13 Standard deviation	24
Equation 14 RMSE	25
Equation 15 MAE	25
Equation 16 Mini-batch mean.....	29
Equation 17 Mini-batch variance	29
Equation 18 Normalization	29
Equation 19 Scaling and shifting	29
Equation 20 L ₂ regularization term.....	30
Equation 21 Cost function with L ₂ regularization	30
Equation 22 Target output discretization	33

List of Tables

Table 1 Performance metrics of existing approach.....	28
Table 2 Experiment results on recurrent batch normalization	34
Table 3 Experiment results of weight decay on LSTM units	35
Table 4 Experiment results of weight decay on perceptron.....	35
Table 5 Experiment results of dropout on LSTM layers	36
Table 6 Experiment results of dropout on perceptron layers	36
Table 7 Experiment results on regularization methods.....	36
Table 8 Experiment results on regularizations.....	37
Table 9 Experiment results on question demoting.....	37
Table 10 Experiment results on optimal model with question demoting	37
Table 11 Experiment results on classification model	38
Table 12 Experiment results on classification model with question demoting	39
Table 13 Experiment results on classification model with regularization and question demoting	39

Chapter 1: Introduction

1.1 Background

One of the main tasks of an educator is to assess the understanding of their students. It can be measured through various assessments, which requires the educator to manually evaluate and grade students' responses. Time taken to do such activity depends on the open-endedness of the assessment since answers to open-ended questions are more varied, and thus, additional time is required to assess the answers. Hence, multiple choice questions will take shorter time to score compared to essay questions. Moreover, scoring open-ended questions is susceptible to the grader's subjectivity as there is no right or wrong answer.

Since most educational system still requires human to manually grade assessments, it consumes a lot of the educators' time. With the help of the automated essay scoring, the grading process would be shortened, and educators would be able to reduce the time they spend evaluating assessments, hence, increasing educators' productivity on other activities. Moreover, the automated essay scoring has numerous advantages over manual scoring, such as its objectivity and consistency.

1.2 Motivation

The education sector would benefit greatly from the automated essay scoring. Despite the fact that multiple choice questions scoring is now mostly done by machine, automated essay scoring is still not widely used even though it has been a research topic for some time now.

One of the existing works on short answer grading is a Siamese Bidirectional LSTM model. This model is claimed to perform better than other approaches, such as the works by Sultan et al. [1], Gomaa et al. [2], and Kumar et al. [3]. However, there is still room for improvement for this approach as the performance is still far from perfect. The author of this model claims that for every score the model predicts, the predicted scores have an average error of 0.618 out of 5, which is relatively high.

1.3 Objectives

The main objective of this project is to improve the grading accuracy of the existing short answer scoring approach, namely the Siamese Bidirectional LSTM model. The approach

will be improved by applying various methods on the model and on the dataset, such as batch normalization, regularization, question demoting, and data augmentation. The methods is targeted to improve the performance of the model, i.e. its Pearson correlation coefficient, RMSE, and MAE.

1.4 Scope

The scope of the questions is limited to short-answered Data Structures questions only so that the answers could be more topic specific. With the limited scope of the questions, the answers will have similar features, and hopefully, the model can pick up important features from the answers more easily.

The model would later on be deployed into a system, and the system would be implemented in an educational setting, where it could help facilitate the scoring process of short-answered data structures related C programming questions, which, in this case, is to assist scoring of CX1007 Data Structures course as a part of students' self-practice and performance evaluation.

Chapter 2: Literature Review

2.1 Past Works

The essay grading issue has been addressed since years ago. One of the earliest approach to the problem is a method known as C-rater [4], which was developed by Educational Testing Service, also widely known as ETS, to measure students' understanding based on their responses to short-answer questions. The method compares the syntactical characteristics of a sentence to a collection of correct ones. However, it is ignoring the difference between passive and active voice, such as “you need two plants” and “two plants are needed”.

More recent work has analysed the difference between corpus-based and knowledge-based measures of text similarity, and it was shown using Pearson's correlation coefficient that corpus-based measure (Lexical Semantic Analysis) performs the best among other approaches. It also introduced new technique which is similar to pseudo-relevance feedback to address the problem where there is more than one correct answer [5].

Although some works use Pearson's correlation coefficient to compare student and reference answers, another experiment [6] showed that similarity measure using Cosine coefficient produce the best result. Cosine similarity measure has the highest accuracy rate compared to other measures, namely Jaccard coefficient and Dice coefficient.

One of the works that use Cosine similarity is the paragraph embeddings [7], which focused on short answer scoring. Answers are considered short “if its length approximately ranges from one phrase to one paragraph”. The word embedding vectors from the answer are combined using average, sum, or other methods, then, using the calculated vectors, new vectors are generated using paragraph embedding model. Cosine coefficient will be used to compare the paragraph vectors.

2.2 Multilayer Perceptron

Multilayer perceptron, a class of feedforward neural network, is a deep learning model that approximates some function by learning parameter so that it could generate the best result. Information flows through the layers of neurons, the first one, being called first layer, until the last one, the output layer. Between the first and output layer, there are more layers, called hidden layers. In contrast to Recurrent Neural Network, information in multilayer perceptron never goes backwards and only goes forward in time [8].

2.3 Batch Normalization

With the discovery of more complex deep learning approaches, the demand of more advanced neural network training techniques became higher. Motivated by the uneven distribution of neural network layers' input during training, batch normalization was introduced in 2015 [9]. The technique is designed to balance the distribution of the inputs by reducing internal covariate shift (ICS), which is the change in the distribution of the input variables in training and test data [10].

2.4 Dropout

Despite the breakthrough of machine learning techniques, overfitting is still a prominent issue in deep learning, especially when the size of training set is very small compared to the complexity of the network. In this case, dropout can be used to avoid the problem. The concept of dropout is to randomly drop neurons from the network, so that dropped neurons will be omitted during training, and hence reducing the complexity of the network [11]. Therefore, the network would be able to learn information that is significant to produce the correct output [12].

2.5 Question Demoting

Another work proposed a new technique named question demoting [13]. The technique removes any words that occur in the question from both the student and gold standard answers. This technique is implemented to eliminate the possibility that students who repeat words from the question in their answers get high score as it does not reflect their understanding in the topic.

2.6 Recurrent Neural Network (RNN)

RNNs are designed to retain information from previous time frames so that the patterns found in the past information can be used to predict the future patterns. Such method is known as long-term dependencies, and RNNs are designed to handle that. However, in practice, RNNs are unable to learn the dependencies, so a new approach, Long Short-Term Memory, was introduced to address this issue [14].

Long Short-Term Memory unit is a type of Recurrent Neural Network. As mentioned in the previous section, the information in this unit goes back in time, as opposed to the traditional multilayer perceptron. It was first introduced in 1997 due to the inability of the conventional approach at that time to prevent the information going backwards to blow up

or vanish [15]. To avoid the aforementioned problem, a forget gate was added to the unit so that insignificant information can be discarded.

The existing work on short answer grading uses this unit on its model as grading a sentence would require understanding each word in the sentence, and the meaning of each word depends on the word preceding it. Hence, the ability of RNNs, or specifically LSTM, to obtain information from the past input is very important in this case.

Chapter 3: Existing Approach

The existing approach automates grading of short answer using Siamese bidirectional LSTM-based regression model. This chapter discusses the deep learning methods implemented in the approach.

3.1 Architecture

This approach combines several neural network architectures, and together they create the overall model architecture. The model consists of several layers, which are shown in Figure 1.

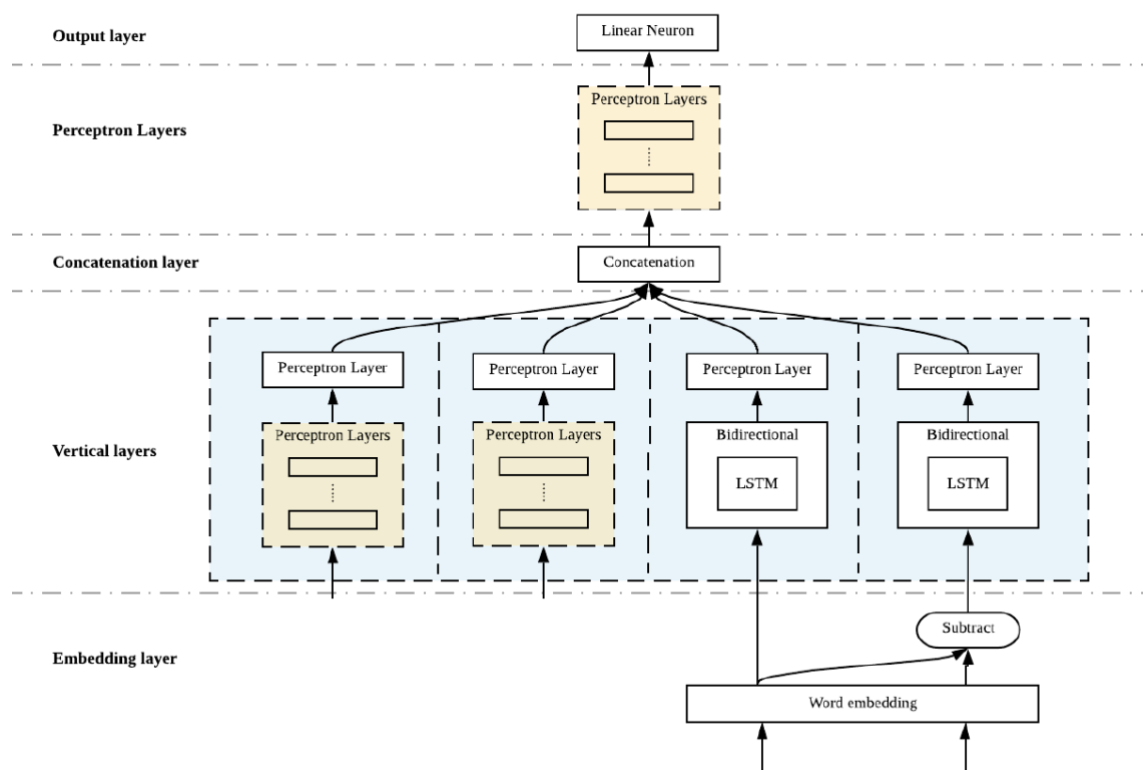


Figure 1 Network architecture illustration

3.1.1 Embedding Layer

The first layer of the network is embedding layer. This layer takes in a sequence of words as input, and each word will be mapped into high-dimensional vectors representing each word, also known as word embedding.

In this layer, Word2Vec is used to compute the vector representations of each word. It is one of word embedding model architectures that utilises Skip-gram model and

negative sampling [16]. The words are represented in such a way that the result of the vector operations reflects the linguistic patterns of the words. For instance, the operation “Madrid” – “Spain” + “France” will produce a vector close to the vector representation of the word “Paris” [17].

The input of this layer will be the answers, while the output will be the mapped 300-dimensional word vectors of the input. This layer will be used twice, for student answers and reference answers.

3.1.2 Vertical Layers

This layer consists of several vertical layers that is independent from each other, and each has its specific functions. Following are the details of the layers as shown in Figure 1 from left to right.

3.1.2.1 Multilayer perceptron (tokenization)

This layer consists of 2 layers of neurons, both of 50 neurons with sigmoid activation function. It receives 3 integers. The first integer is number of words of the reference answer, the second is number of the student answer, and the last one is number of words that exist both in the student and reference answers. To obtain these numbers, the answers are first tokenized using NLTK word tokenizer. Tokenization is a process where a sequence of words is separated into smaller parts called tokens. [18].

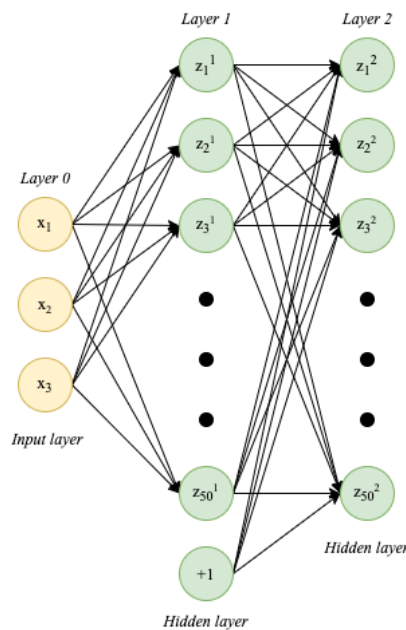


Figure 2 Multilayer perceptron (tokenization)

3.1.2.2 Multilayer perceptron (feature engineering)

This layer takes input of 5 set of data obtained from feature engineering. Feature engineering is a process where a certain data is produced by transforming raw data so that it could be used by the model to better learn the pattern of the given data [19].

In this case, length of the student answer (number of characters), ratio of the length of the reference answer and length of the student answer, number of words in the student answer, and the number of unique words in the student answer are used.

The input is then processed through 4 layers of 125 neurons with sigmoid activation function.

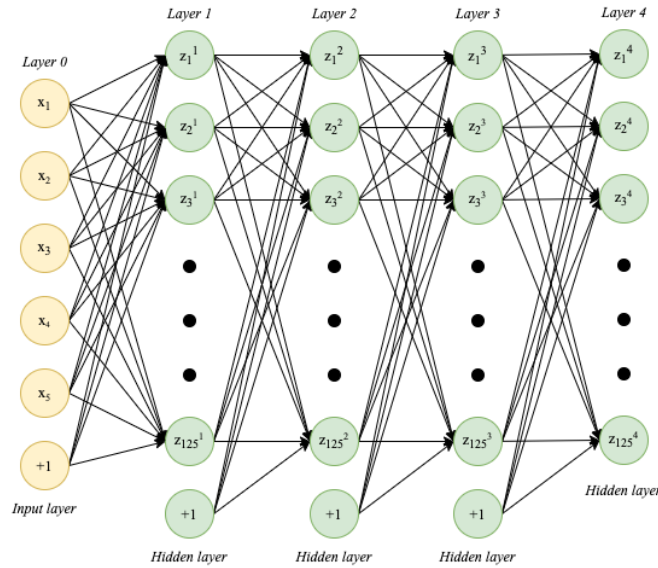


Figure 3 Multilayer perceptron (feature engineering)

3.1.2.3 LSTM unit (reference answer)

This layer receives input from embedding layer, which is the embedded student answers. It will then be propagated through a bidirectional LSTM unit and through a layer of 50 neurons with sigmoid activation function. The LSTM unit will be discussed further in Section **Error! Reference source not found.**

3.1.2.4 LSTM unit (comparison with student answer)

This layer has the same architecture as the previous layer, but it has different inputs. Instead of receiving input straight from the embedding layer, this

layer receives a vector of comparison between student answer and reference answer. The two sentences are compared using the distance of the two vectors. The distance is calculated using the equation below, where R_i is the reference answer of question Q_i , and A_{ij} is the student answer j of question Q_i . $v(X)$ is the word vector of sentence X .

$$dist = v(R_i) - v(A_{ij})$$

Equation 1 Distance between two vectors

Word2Vec groups similar words together [17], so words with similar meaning has similar vector values. The distance of the two vectors should be small if the sentences have similar meaning. Therefore, cosine coefficient is not used to calculate the similarity of the two words since the word embedding has this feature even though Pribadi et al. [6] claims that it is the most accurate similarity measure.

3.1.3 Concatenation Layer

In this layer, the output from vertical layers are merged to be processed further in the following layers.

3.1.4 Perceptron Layers

The concatenated vector is then propagated through a multilayer perceptron. The multilayer perceptron has 4 layers, where the first 3 layers consist of 125 neurons, and the last one consists of 25 neurons.

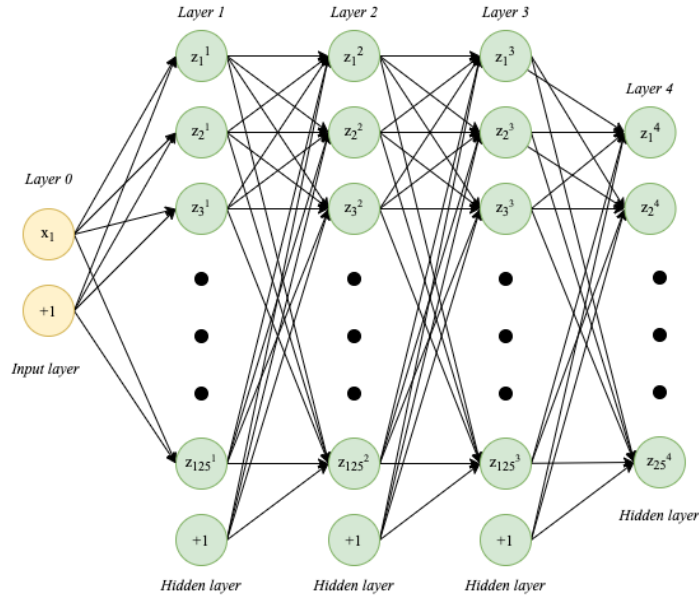


Figure 4 Perceptron layers

3.1.5 Output Layer

The final layer of the network consists of 1 neuron with linear activation function. This neuron outputs the final predicted score.

3.2 Bidirectional LSTM Unit

The basic LSTM unit consists of a cell state that modulates information through the unit, and three gates (forget, input, and output) [14]. Following is the functions used in the unit, where $x(t)$ is the input and $h(t)$ the output.

3.2.1 Forget gate

Forget gate will determine whether the information should be removed or not. When it is 0, nothing will go through.

$$f(t) = \sigma(U_f x(t) + W_i h(t-1) + b_f)$$

Equation 2 Forget gate

Where $\sigma(x)$ is sigmoid function.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Equation 3 Sigmoid function

3.2.2 Input gate

3.2.2.1 Sigmoid layer

This layer will determine which values to update.

$$i(t) = \sigma(U_i x(t) + W_i h(t-1) + b_i)$$

Equation 4 Input gate sigmoid layer

3.2.2.2 Tanh layer

This layer will produce vector of new candidate values.

$$\tilde{C}(t) = \phi(U_c x(t) + W_c h(t-1) + b_c)$$

Equation 5 Input gate tanh layer

Where $\phi(x)$ is tanh function.

$$\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Equation 6 Tanh function

3.2.3 Output gate

This gate will allow memory cell to have effect on other neurons.

$$o(t) = \sigma(U_o x(t) + W_o h(t-1) + b_o)$$

Equation 7 Output gate

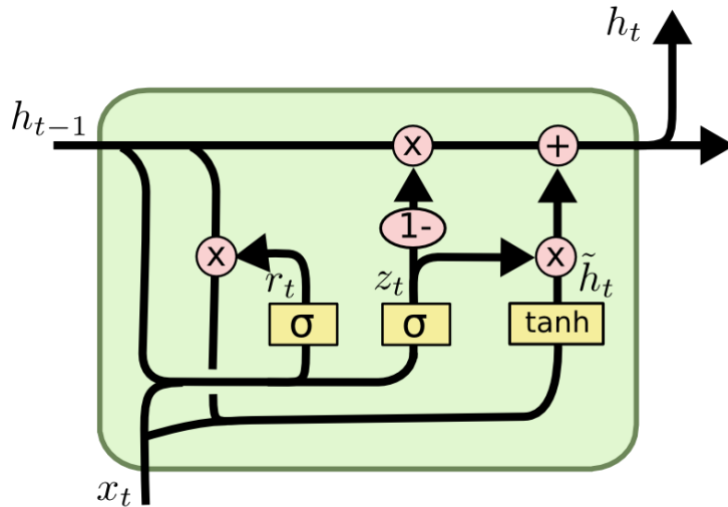


Figure 5 LSTM Unit

Afterwards, the cell state is updated using equation below, where \odot is element-wise product.

$$C(t) = \tilde{C}(t) \odot i(t) + C(t-1) \odot f(t)$$

Equation 8 Cell state

The output value is calculated using the equation below.

$$h(t) = \phi(C(t) \odot o(t))$$

Equation 9 LSTM output

However, this approach uses bidirectional LSTM, which means the neurons are split into two directions, forward and backward. This method will enable the effective usage of both past and present information for a specific time frame [20].

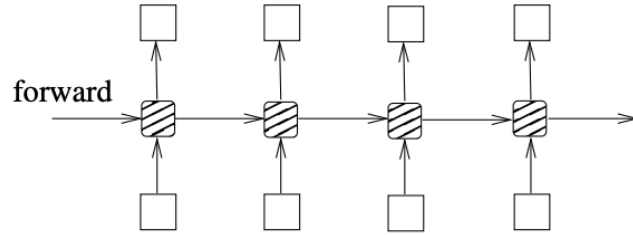


Figure 6 LSTM network

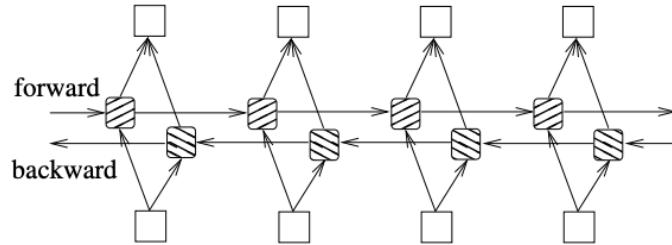


Figure 7 Bidirectional LSTM network

Two bidirectional LSTM units with identical architecture are used for this approach. Hence, this model is also called a Siamese Bidirectional LSTM.

Chapter 4: Experimental Evaluation

4.1 Dataset

This experimental evaluation uses a dataset from an experiment conducted by Mohler, Bunescu, and Mihalcea [13].

The dataset consists of data structures questions for introductory computer science assignment at the University of North Texas. There are total of 80 questions and 31 students enrolled in the course. In total, the dataset consists of total 2273 student answers since some students did not submit any answer for some questions.

Each answer is graded manually by two human graders, and the score is an integer ranging from 0 to 5, where 5 indicates a perfect answer. The average of the two scores is then used as gold standard of this experiment.

4.2 Data Pre-Processing

Firstly, data cleaning is performed on the dataset. It is cleaned by removing tuples with empty values. All tuples with empty question, answer, or scores are removed. Then, all the questions and answers are converted into strings, and stopwords and punctuations are removed from the sentence. Next, the scores are scaled using min-max normalization so that it ranges from 0 to 1.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Equation 10 Min-max normalization

4.3 Evaluation Metrics

4.3.1 Pearson Correlation Coefficient

Pearson correlation coefficient measures the linear correlation of two variables. The coefficient ranges from -1 to 1, where 1 means the two variables are positively correlated and -1 negatively correlated.

Pearson correlation coefficient ρ of two variables X and Y is defined as below.

$$\rho_{X,Y} = \frac{S_{X,Y}}{S_X S_Y}$$

Equation 11 Pearson correlation coefficient

$$s_{X,Y} = \frac{1}{n-1} \sum_{k=1}^n (X_k - \bar{X})(Y_k - \bar{Y})$$

Equation 12 Covariance

$$s_X = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (X_k - \bar{X})^2}$$

Equation 13 Standard deviation

Figure 8 below shows the correlation between two variables X in X-axis and Y in Y-axis.

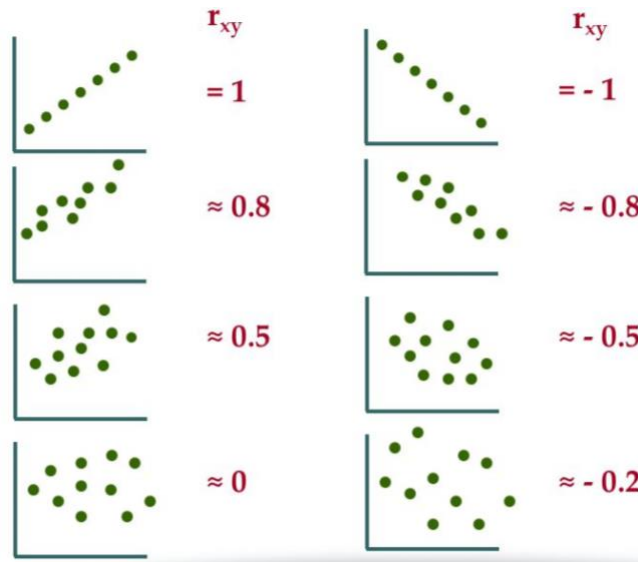


Figure 8 Pearson correlation coefficient

Pearson correlation coefficient is calculated to evaluate the approach. However, it only represents linear correlation of the two vectors and does not represent the error or how much the vector deviate from each other. For example, vectors [3,4,5] and [1,2,3] will have the coefficient equals to 1. Therefore, this metric cannot be used alone to determine the performance of a model.

4.3.2 Rooted Mean Square Error

Rooted mean square (RMSE) is the square root of the arithmetic mean of the squares of the difference between the two variables.

$$RMSE = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (y_k - \hat{y}_k)^2}$$

Equation 14 RMSE

In this experiment, $y_{1...k}$ are target gold standard values, and \hat{y}_k are predicted values by the model. Since this metric represents the difference between the two values, this metric can be used to observe the performance of a model. Lower RMSE means the model could predict values that are closer to the given dataset.

For example, the RMSE of an error of 0.5 in grading would be 0.25, while an error of 1 in grading would result be 1. From this example, it can be concluded that even though the RMSE is 4 times as large, does not mean that it is 4 times as bad.

4.3.3 Mean Absolute Error

Mean absolute error (MAE) measures the difference of two variables. It is the arithmetic mean of the absolute difference of the two variables.

$$MAE = \frac{1}{n-1} \sum_{k=1}^n |y_k - \hat{y}_k|$$

Equation 15 MAE

$y_{1...k}$ are observed values and \hat{y}_k are predicted values. This metric also represents the difference between the two values and can be used to evaluate model performance. Lower MAE also means that the model could predict values close to the actual data.

Using similar example from the previous section, an error of 0.5 in grading would have MAE 0.5, and error of 1 would result in 1. In this case, it can be assumed that the error is twice as bad. Due to this property, MAE is considered the best measure to compare different models' performance in this experiment and is used as the loss function during the training of the model.

4.4 Data Augmentation

The dataset only consists of 2273 tuples. This number of data is not sufficient to train a complex network, and it will cause the model to overfit the data. Hence, to avoid this issue, data augmentation is applied to obtain larger dataset.

Data augmentation is a method to increase the size and diversity of the dataset for training without actually collecting new data, so the new data is obtained from the dataset itself [21].

In this experiment, student answers that are given perfect score 5.0 by human graders are used as new reference answer. Thus, if n out of m students received perfect grade in a particular question, $(n - 1) * m$ new data tuples can be generated for that question. By performing this technique, around 30000 data samples were generated and used to train the model.

4.5 Evaluation

4.5.1 Implementation

The model was built using Keras, a high-level neural networks API that is able to run on top of TensorFlow [22]. The code is written on Python and implemented with the help of other machine learning libraries mentioned below.

4.5.1.1 *pandas*

pandas is a data analysis and manipulation tool [23]. This library is used in this experiment to manipulate and process the data so that it can be used to train the model.

4.5.1.2 *NumPy*

NumPy is a library that supports large dimensional array operations [24]. This library is used in this experiment since Keras requires its input to be in *NumPy* array format.

4.5.1.3 *scikit-learn*

scikit-learn is a machine learning library that supports tools for predictive data analysis. It provides numerous machine learning features such as classification, regression, clustering [25]. In this experiment, its model selection, metrics, and pre-processing built-in functions are used.

4.5.1.4 *Natural Language Toolkit (NLTK)*

NLTK is a natural language processing library [26]. It provides list of English stopwords and tokenization, which are used in this experiment.

4.5.2 Model Selection

The model selection method used was a combination of three-way data splits and 5-fold cross validation methods.

The dataset was first split into training and test set on 80:20 ratio. The 80% of the data is divided into 5-folds of training and validation dataset. Then, the model is trained using 5-fold cross validation method, and the best model is chosen based on the average of RMSE and MAE on the validation set. The fold that has lowest average of RMSE and MAE is selected. Lastly, the chosen model is used to predict values from unseen data (test set) and the result from test set is used to evaluate the overall model performance.

The data split ratio 80:20 is chosen upon some considerations. The training set, which is used to train the model, might be too small if 70:30 ratio is used, considering the complexity of the model. On the other hand, 90:10 ratio will produce very small test set. Since test set will be used to assess the model performance, if it is too small, it may not correctly represent the performance.

4.5.3 Training

For each fold on training set, the model is trained on GPU using 150 epochs since the loss seems to be stable around that iteration. The parameter is updated using mini-batch stochastic gradient descent algorithm, specifically AdaGrad optimization method, and the loss function is mean absolute error (MAE) as stated in Equation 15.

AdaGrad is used because it is not sensitive to hyperparameters as it is dynamic and adaptable to the data and generate different learning rates for different features. Parameters of features that occur frequently will be applied lower learning rates than of features that occur infrequently [27]. The input of the model used in this experiment is diverse, and hence, using AdaGrad as the optimization method will help the model to train better.

4.6 Performance of Existing Model

Using the evaluation techniques explained in the previous section, the existing model performance is as below.

Pearson coefficient	RMSE	MAE	Avg(RMSE+MAE)
0.4416	0.9815	0.6471	0.8143

Table 1 Performance metrics of existing approach

From the metrics above, it is shown that the model does not perform very well as the Pearson correlation coefficient is less than 0.5. The low coefficient means that the predicted values are not linearly correlated to the target values.

Most importantly, its MAE is fairly high, which is 0.6471. It means that for all gold standard score, the average difference with the predicted score is 0.6471 out of 5.0, which the error is 12.94% of the total score. For instance, if the gold standard score is 4, the model would probably predict the score to be 4.6471, which is closer to 5 than to 4.

Below is the training and test error against the number of iterations.

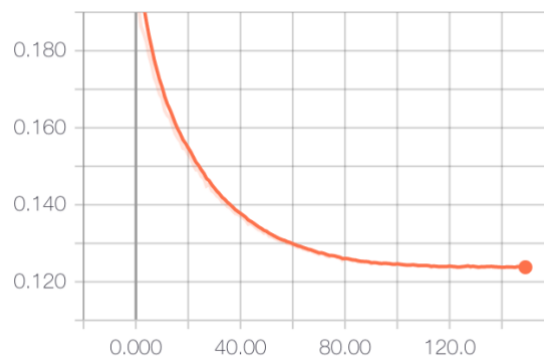


Figure 9 Training error of original model

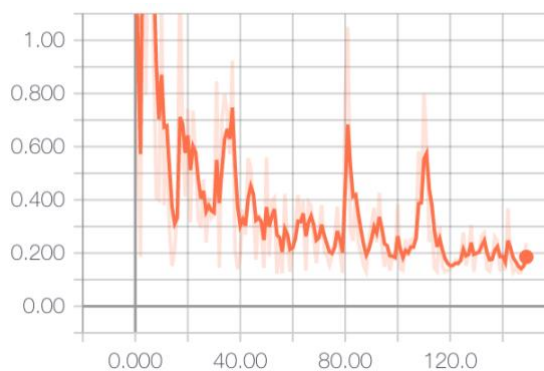


Figure 10 Validation error of original model

As we can see from the graphs above, the graph of the validation error against number of iteration seems to be very bumpy and spiky even though the graph of training error is very smooth. It means that the model does not perform well on the unseen data and it is

overfitting on the training dataset. As a result, the model is unable to predict accurately on unseen data and produce high error on validation dataset.

4.7 Improving Model Performance

Since the original model is overfitting on the training data, the model should be improved to result on better performance. There are many techniques that can be implemented to improve the model performance.

4.7.1 Batch Normalization

The input and output of batch normalization layer are four dimensional vectors of batch, channel, and two spatial dimensions [28]. It normalizes each input of a layer by subtracting mini-batch mean and dividing it by the mini-batch standard deviation. However, it may change the input's representation [10].

For input $x = \{x_{1...m}\}$ and output y_i over a mini-batch B, batch normalization needs to learn parameters γ and β , where ϵ is a constant added for numerical stability.

$$\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^m x_i$$

Equation 16 Mini-batch mean

$$\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$$

Equation 17 Mini-batch variance

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

Equation 18 Normalization

$$y_i \leftarrow \gamma \hat{x}_i + \beta$$

Equation 19 Scaling and shifting

The original model only performs batch normalization on the last perceptron layer. In this experiment, additional batch normalization is applied on each of the LSTM layers.

4.7.2 Regularization

Regularization is a strategy to reduce error on unseen data or test error by altering the learning algorithm [8]. Even though this could increase the training error, reducing test error would improve the overall performance of a model since with lower test error, the model would predict more accurately on unseen data.

4.7.2.1 Weight regularization

Weight regularization is done by adding penalty or regularization term on the cost or loss function. The most commonly used weight regularization is L2 regularization, also known as ridge penalization. This approach uses L2 regularization term, which is defined below [29].

$$L_2 \text{ regularization term} = w_1^2 + w_2^2 + \dots + w_n^2$$

Equation 20 L2 regularization term

This term is added to the cost function J . Since the goal of the algorithm is to minimize the cost function, when the penalization β_2 is increased, the weights w_{ij} would be decreased.

$$J \leftarrow J + \beta_2 \sum_{i,j} (w_{ij})^2$$

Equation 21 Cost function with L2 regularization

When the weights have smaller values, the model would be simpler and, as a result, less prone to overfitting.

In this experiment, different values of parameter β_2 will be used to evaluate the optimal parameter.

4.7.2.2 Dropout

Dropout is a method to avoid overfitting by omitting some neurons in the network.

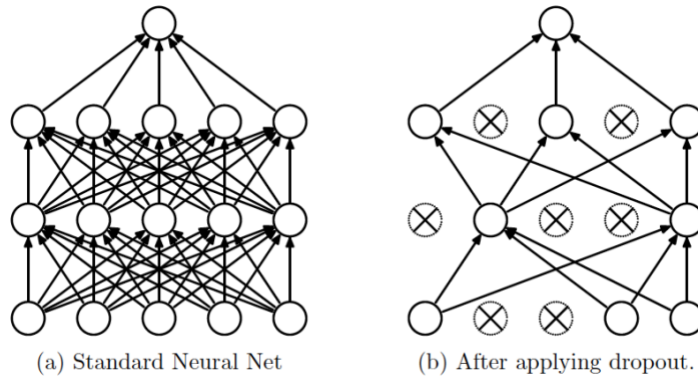


Figure 11 Dropout on neural network with 2 hidden layers

By performing dropout, random neurons, which the number of them is determined by the dropout rate, are omitted. Hence, the model becomes less complex and it can better learn important features from the dataset [11].

In the original model, the neurons in perceptron layer is dropped out 50%. This experiment will try to perform dropout on the LSTM layer and try different dropout rates on the perceptron layer.

4.7.3 Question Demoting

During data pre-processing, all words in reference and student answers that are in the question are removed. However, there are some cases where the answer is in the question sentence. For example, the question “What data structure is more appropriate for scheduling printing jobs at a printer, a stack or a queue?”. The answer of that question would be either “Stack” or “Queue”. When question demoting is performed on this question and its answers, the answers will be empty. Hence, to avoid this issue, the question is altered by removing the options, i.e. “a stack or a queue”.

4.7.4 Classification Model

The model is converted into a classification model with expectation that it results in relatively better performance.

4.7.4.1 Motivation

The answers from the dataset is scored using ordinal values, which are 0, 1, 2, 3, 4, and 5, but there is no rubric on how to score the answers, so each

grader may have different opinion on how an answer should be graded. For example, one grader might think an answer should be graded 4, while the other think it should be 3. It is found that human grader only agrees on a score only 53% to 81% of the time [30].

Moreover, the model that is used to predict the score is a regression model. The predicted answer would be a continuous number, and since there is no rubric on how to score the answer, there is no way to determine how accurate an answer is based on the score. For example, when an answer is graded 5 and another answer is graded 4.8, it is difficult to show how one is better than another since the difference is so small and the answer is so short that there are not many features that can be evaluated to grade the answer.

4.7.4.2 *Implementation*

The model is modified into a multi-category classification model, where the output is three classes, 0, 1, and 2. An answer is labelled 0 or ‘wrong’ when it is completely wrong or completely different from the reference answer. An answer is labelled 1 when it is only partially correct or partially wrong. Lastly, an answer is labelled 2 or ‘correct’ if it is a perfect answer and has the same connotation as the reference answer.

The model is built by modifying the output layer, the very last layer, of the original model into a softmax layer with 3 outputs. Softmax function is a function that is commonly used in the last layer of categorical model. It represents the probability distribution over possible discrete variable values [8]. The variable that has the highest probability will be the predicted output of the model.

The model is trained with the dataset, where an additional pre-processing is applied. The target output, which is the average scores of two graders x , are discretized into 3 categories using the function below.

$$f(x) = \begin{cases} 0, & x \leq 1 \\ 1, & 1 < x < 4 \\ 2, & x \geq 4 \end{cases}$$

The scores are discretized into 3 values since 5 values might still be too broad and there is no objective justification on how one can distinguish how an answer with score 4 is better than one that is scored 3.

4.7.4.3 Evaluation metrics

Evaluation metrics used to evaluate the classification model is different from regression model as the output is discrete values. Accuracy, precision, recall, and F1 score are used to evaluate the model.

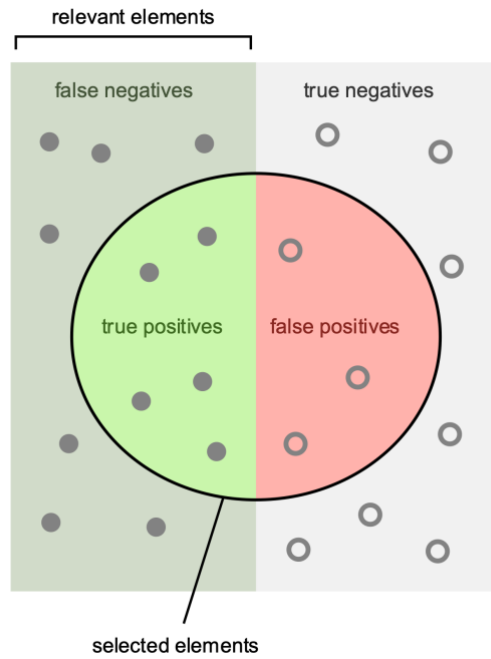


Figure 12 Classifier terms

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy is the percentage of data that are correctly predicted by the model. However, accuracy is often misleading since it is imbalanced. A dataset with 95 positive data and 5 negative data will still have accuracy of 95% although it predicts all data to be positive [31].

$$Precision = \frac{TP}{TP + FP}$$

Precision is the ratio of data of a class that is correctly predicted over the size of data that is predicted to be in that class, but it is biased by the false positive

$$Recall = \frac{TP}{TP + FN}$$

Recall is the ratio of data of a class that is correctly predicted over the size of data that is actually in that class, and it is biased towards the false negative.

$$F1\ score = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

F1 score was introduced to address the issues found in recall, precision, and accuracy.

4.8 Experiment Results

This section will discuss the performance of the model when improvement techniques above are applied. The optimal values in the tables below are in bold, and complete results of the experiment on regression model can be found on Appendix A: Experiment Results (Regression Model).

4.8.1 Recurrent Batch Normalization Results

Model	Pearson	RMSE	MAE	Avg(RMSE,MAE)
Original	0.4416	0.9815	0.6471	0.8143
Recurrent BN	0.4426	0.9648	0.6429	0.8039

Table 2 Experiment results on recurrent batch normalization

Applying batch normalization on the recurrent network improves the model performance for all metrics since it helps the model to perform better on test data and the weights of the model are normalized.

4.8.2 Regularization Results

Several experiments on regularization were conducted on different parts of the model. Different values of parameter are also experimented to obtain the best result of the regularization.

4.8.2.1 Weight regularization on LSTM units

Experiments on different parameters of the L2 regularization is conducted. All experiments have same values for bias, weights, and recurrent parameters, as stated on the table below.

Parameter	Pearson	RMSE	MAE	Avg(RMSE,MAE)
0.01	0.4512	0.9787	0.6293	0.8040
0.001	0.4526	0.9556	0.6466	0.8011
0.0001	0.4513	0.9563	0.6467	0.8015

Table 3 Experiment results of weight decay on LSTM units

The result shows that parameter 0.001 has the best performance among other parameters on all metrics.

4.8.2.2 Weight regularization on perceptron

Weight decay was applied to the last 2 perceptron layers on the 2 multi-layer perceptron on vertical layer and perceptron layer (refer to Figure 1).

Values for bias and weights parameters are the same, as stated below.

Parameter	Pearson	RMSE	MAE	Avg(RMSE,MAE)
0.01	0.4471	0.9694	0.6309	0.8002
0.001	0.4491	0.9677	0.6571	0.8124
0.0001	0.4504	0.9952	0.6349	0.8151

Table 4 Experiment results of weight decay on perceptron

Weight decay with bias and weight parameter 0.01 results on model with the best performance. Even though parameter 0.0001 has the highest Pearson correlated coefficient, as elaborated on Section 4.3.1, it cannot be used independently to evaluate model performance.

4.8.2.3 Dropout on LSTM

Dropout with different rates are applied to the model on the LSTM layers.

Rate	Pearson	RMSE	MAE	Avg(RMSE,MAE)
0.2	0.4476	0.9697	0.6445	0.8071
0.4	0.4506	0.9632	0.6432	0.8032

0.5	0.4413	0.9599	0.6629	0.8114
-----	--------	--------	--------	--------

Table 5 Experiment results of dropout on LSTM layers

From the experiment, it is found that dropout rate 0.4 has the best result among other rates.

4.8.2.4 Dropout on perceptron

On the original model, dropout rate 0.5 is used on the perceptron layers. To find the optimal rate, lower rate is first used to see if it improves the model. Then, after seeing that lower rate does not improve the performance, higher rate is used.

Rate	Pearson	RMSE	MAE	Avg(RMSE,MAE)
0.2	0.4436	1.0034	0.6369	0.8202
0.5	0.4416	0.9815	0.6471	0.8143
0.6	0.4504	0.9778	0.6450	0.8114

Table 6 Experiment results of dropout on perceptron layers

Dropout with rate 0.6 has the best overall performance even though its MAE is still higher than model with rate 0.2.

All regularization methods are combined, and based on the experiments above, the optimal parameter values are used.

Model	Pearson	RMSE	MAE	Avg(RMSE,MAE)
Original	0.4416	0.9815	0.6471	0.8143
Regularized	0.4486	1.0199	0.6294	0.8247

Table 7 Experiment results on regularization methods

However, the performance decrease as the model seem to be underfitting the data. Hence, dropout rate is reduced back to 0.5, and afterwards, recurrent batch normalization is implemented along with all the regularization methods.

Model	Pearson	RMSE	MAE	Avg(RMSE,MAE)
Original	0.4416	0.9815	0.6471	0.8143
Regularization methods	0.4494	0.9504	0.6523	0.8014

Regularization and recurrent batch normalization	0.4461	1.0000	0.6200	0.8100
--	--------	--------	---------------	--------

Table 8 Experiment results on regularizations

The overall performance is better after the dropout rate is reduced. Then, recurrent batch normalization is added to the model. However, the performance does not seem to be improved.

4.8.3 Question Demoting

Question demoting is applied to the dataset, and the original model is trained using the demoted answers.

Model	Pearson	RMSE	MAE	Avg(RMSE,MAE)
Original	0.4416	0.9815	0.6471	0.8143
Question demoting	0.469	0.9761	0.6204	0.7983

Table 9 Experiment results on question demoting

The performance improved quite significantly after question demoting is applied. Hence, question demoting is performed along with the optimal models from the previous section.

Model	Pearson	RMSE	MAE	Avg(RMSE,MAE)
Original	0.4416	0.9815	0.6471	0.8143
Regularized	0.4639	0.9604	0.6124	0.7864
Regularized and BN	0.4601	0.9816	0.6091	0.7954

Table 10 Experiment results on optimal model with question demoting

The model without recurrent batch normalization is shown to perform better as the overall metrics are better, and among other improvement, this method has the best performance.

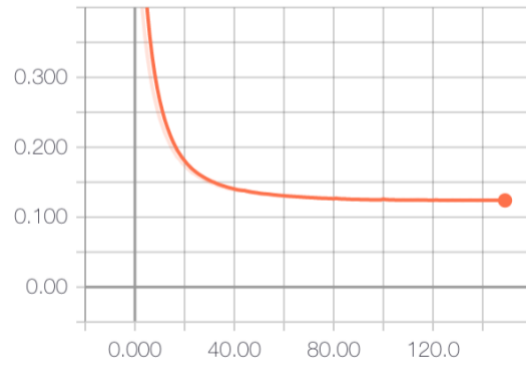


Figure 13 Training loss of optimal model

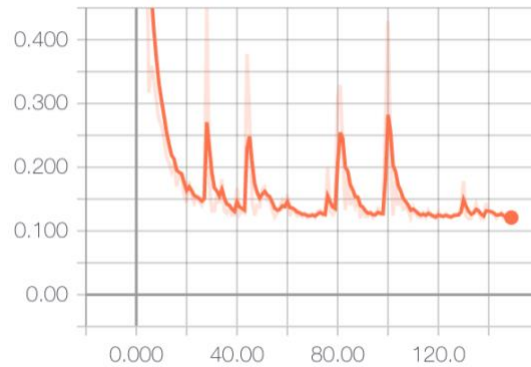


Figure 14 Validation loss of optimal model

Compared to the original model, the performance of the model after improvement is relatively better, especially on unseen test data. The validation loss is significantly lower and more stable than the one from the original model (Figure 10).

4.8.4 Classification Results

Using the parameters from the previous sections, the classification model is trained using dataset has been question demoted.

4.8.4.1 Original model

Class	Precision	Recall	F1 Score	Support
0	0.00%	0.00%	0.00%	92
1	55.53%	21.01%	30.49%	1266
2	80.68%	95.68%	87.54%	4726
Weighted avg	74.23%	78.70%	74.35%	
Accuracy	78.70%			

Table 11 Experiment results on classification model

4.8.4.2 Original model with question demoting

Class	Precision	Recall	F1 Score	Support
0	31.13%	51.09%	38.68%	92
1	51.12%	36.05%	38.32%	1266
2	83.09%	90.96%	86.85%	4726
Weighted avg	75.65%	77.81%	76.02%	
Accuracy	77.81%			

Table 12 Experiment results on classification model with question demoting

4.8.4.3 Model with regularization and question demoting

Class	Precision	Recall	F1 Score	Support
0	0.00%	0.00%	0.00%	92
1	47.96%	19.51%	27.73%	1266
2	80.97%	95.41%	87.60%	4726
Weighted avg	72.87%	78.17%	73.82%	
Accuracy	78.17%			

Table 13 Experiment results on classification model with regularization and question demoting

Based on the results above, it can be concluded that applying regularization and question demoting does not improve the performance of the classification model as the accuracy is lower.

Chapter 5: System Implementation

5.1 Software Architecture

The system is implemented using Django framework, and its interface is built using React library.

5.1.1 Design Pattern

Django is an open-source, high-level Python web framework that supports rapid development and clean, pragmatic design [32]. It follows the Model-View-Template architecture pattern, also known as MVT. The abstraction layer, model, provides structuring and manipulates the data. The view layer is responsible for the logical operations and for processing user requests, including returning the responses. Lastly, template layer renders the information to be presented to the user [33].

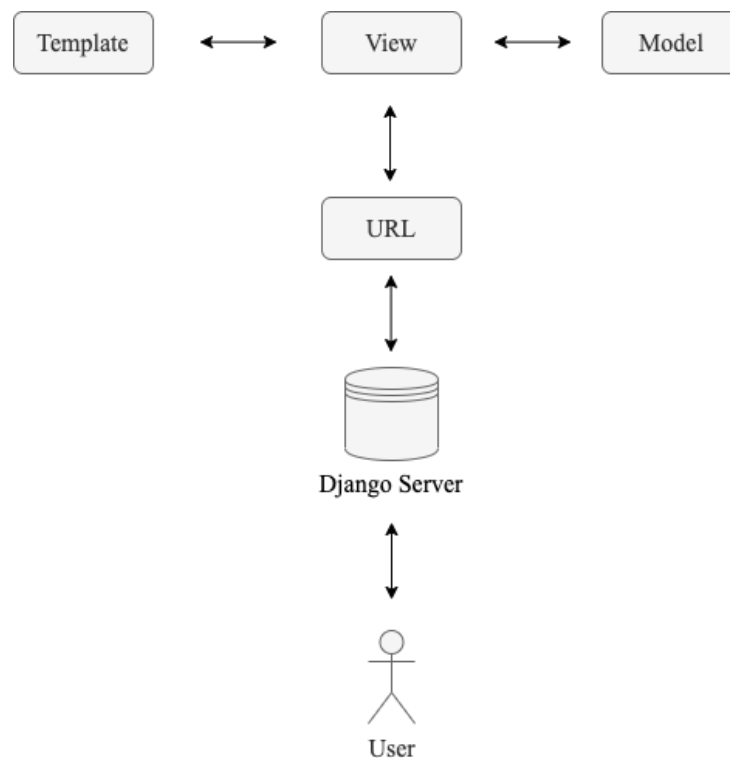


Figure 15 Django architecture pattern

5.1.2 Frameworks

In this system, the template layer is implemented with the support of Django REST Framework and React library. Django REST Framework, which stands for Representational State Transfer, is a powerful and flexible toolkit for building Web

APIs [25]. It helps facilitate the usage of Django Server and acts as a RESTful API. Information from Django Server is passed through Django REST Framework to React. Then, users will be able to view the information from the database in Django server through the interface that is provided by React App. React app will get information from Django database using POST, GET, and PUT requests.

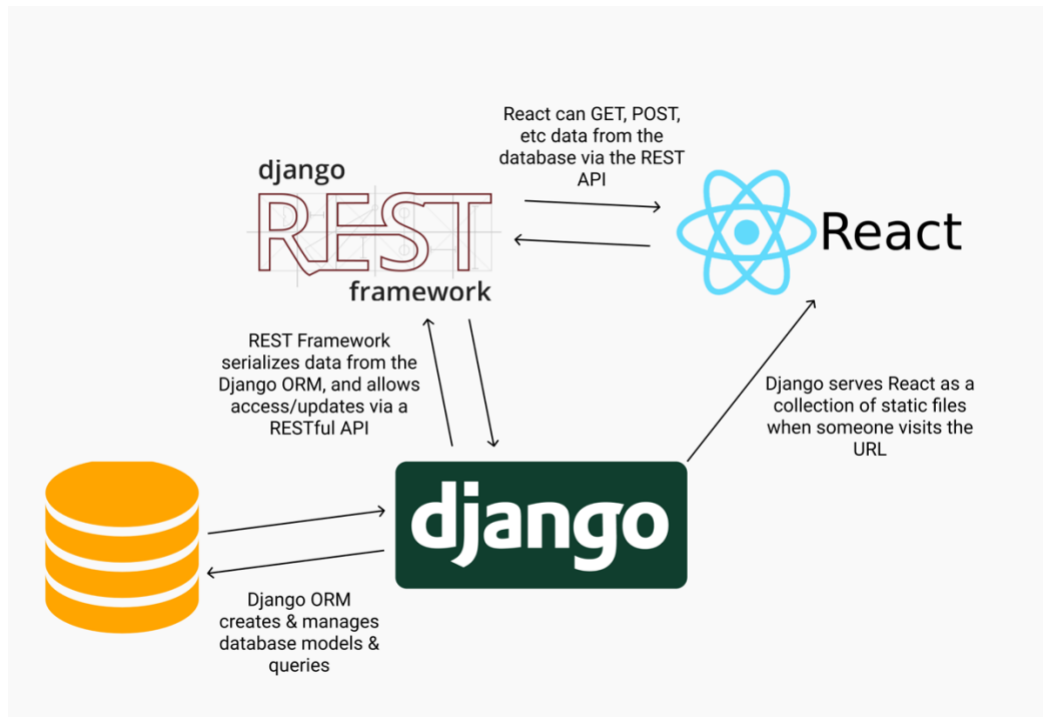


Figure 16 Django and React with Django REST Framework

5.1.3 Database

The system uses the default database provided by Django, which is SQLite. It supports small, fast, self-contained, high reliability, full-features, SQL database engines [34]. Therefore, access to the database is easier since Django provides built-in functions for communication with SQLite database.

5.2 System Interface

A prototype system implementation has been built using the architecture mentioned in the section above. This system is built with the intention to collect more data on conceptual short-answered data structure questions, and, at the same time, serving as a platform from students to practice. Data obtained from this system can be used to train the LSTM network, and, hopefully, the model performance could be improved when trained using the additional data.

5.2.1 General

This section covers pages in the website that is accessible for users with no account.

5.2.1.1 Sign up

Prior to using the system, everyone has to create an account in order to access the website. In this page, user can choose which type of user to sign up, admin or student. The features available for each type will be discussed in their respective sections, section 5.2.2 and section 5.2.3.

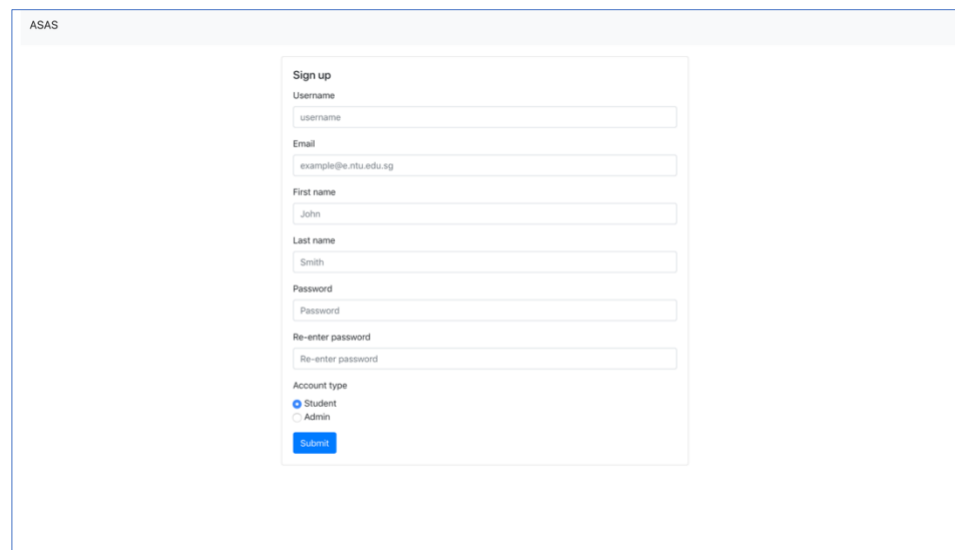
The image shows a web browser window with the title 'ASAS'. Inside the window is a 'Sign up' form. The form contains the following fields: 'Username' (with placeholder text 'username'), 'Email' (with placeholder text 'example@e.ntu.edu.sg'), 'First name' (with placeholder text 'John'), 'Last name' (with placeholder text 'Smith'), 'Password' (with placeholder text 'Password'), and 'Re-enter password' (with placeholder text 'Re-enter password'). Below these fields are two radio buttons for 'Account type': 'Student' (which is selected) and 'Admin'. At the bottom of the form is a blue 'Submit' button.

Figure 17 Sign up page

5.2.1.2 Log in

Once the account is approved, user can sign in as the respective type of user they signed up for. This page is the home page of the website when a user is not yet logged in.

Figure 18 Log in page

5.2.2 Admin

Admin users in this website will be able to upload questions and score student answers. These scores will be used as gold standard during training process.

5.2.2.1 Questions

Admin can view all questions in the database from this page. The reference answer of the respective question is also displayed in this page.

ASAS	QUESTIONS	ANSWERS	ADD QUESTIONS	ADD ANSWERS	POSTS	MODEL	STUDENTS	LOGOUT
ID	Question	Reference answer	Action					
23	What is the role of a prototype program in problem solving?	To simulate the behaviour of portions of the desired software product.	Details					
24	What stages in the software life cycle are influenced by the testing stage?	The testing stage can influence both the coding stage (phase 5) and the solution refinement stage (phase 7)	Details					
25	What are the main advantages associated with object-oriented programming?	Abstraction and reusability.	Details					
26	Where do C++ programs begin to execute?	At the main function.	Details					
27	What is a variable?	A location in memory that can store a value.	Details					
28	Where are variables declared in a C++ program?	Variables can be declared anywhere in a program. They can be declared inside a function (local variables) or outside the functions (global variables)	Details					
29	What is the main difference between a while and a do...while statement?	The block inside a do...while statement will execute at least once.	Details					
30	What is typically included in a class definition?	Data members (attributes) and member functions.	Details					
31	What is the difference between a data member and a local variable inside a member function?	Data members can be accessed from any member functions inside the class definition. Local variables can only be accessed inside the member function that defines them.	Details					
32	What is the difference between a constructor and a function?	A constructor is called whenever an object is created	Details					
33	When does C++ create a default constructor?	If no constructor is provided	Details					
34	How many constructors can be created for a class?	Unlimited number.	Details					
35	What is the difference between a function prototype and a function definition?	A function prototype includes the function signature	Details					

Figure 19 Admin questions list

By clicking “Details” button on the right column of the table, user can view all answers of that particular question submitted by any student. Scores of the answer is also displayed. System score column will show the scores predicted by the neural network.

Admin can score the answers through this page by clicking “Score” button on top right. More answers for this particular question can also be added automatically by uploading a CSV file that contains answers and their scores.

ASAS QUESTIONS ANSWERS ADD QUESTIONS ADD ANSWERS POSTS MODEL STUDENTS LOGOUT

Question

What is the role of a prototype program in problem solving?

Reference answer

To simulate the behaviour of portions of the desired software product.

Back

Score

Add from CSV

Choose file

N..n

ID	Student answer	System score	Score 1	Score 2
332	High risk problems are address in the prototype program to make sure that the program is feasible. A prototype may also be used to show a company that the software can be possibly programmed. 	-	3.5	3.5
333	To simulate portions of the desired final product with a quick and easy program that does a small specific job. It is a way to help see what the problem is and how you may solve it in the final project.	-	5	5
334	A prototype program simulates the behaviors of portions of the desired software product to allow for error checking.	-	4	4
335	Defined in the Specification phase a prototype stimulates the behavior of portions of the desired software product. Meaning, the role of a prototype is a temporary solution until the program itself is refined to be used extensively in problem solving.	-	5	5
337	It is used to let the users have a first idea of the completed program and allow the clients to evaluate the program. This can generate much feedback including software specifications and project estimations of the total project.	-	3	3
338	To find problem and errors in a program before it is finalized	-	2	2
339	To address major issues in the creation of the program. There is no way to account for all possible bugs in the program, but it is possible to prove the program is tangible.	-	2.5	2.5
341	you can break the whole program into prototype programs to simulate parts of the final program	-	5	5
342	-To provide an example or model of how the finished program should perform. -Provides foresight of some of the challenges that would be encountered. -Provides opportunity To introduce changes To the finished program.	-	3.5	3.5
343	Simulating the behavior of only a portion of the desired software product.	-	5	5
344	A program that stimulates the behavior of portions of the desired software product.	-	5	5

Figure 20 Admin question details

To upload new questions to the system, admins can upload by typing down the questions manually and the reference answer, where the questions will be added one by one, or by uploading a CSV file, where multiple questions can be added simultaneously.

ASAS QUESTIONS ANSWERS ADD QUESTIONS ADD ANSWERS POSTS MODEL STUDENTS LOGOUT	
<p>Manual</p> <p>Add question to the database manually.</p> <p>Question</p> <input type="text"/> <p>Reference answer</p> <input type="text"/> <p>Add question</p>	<p>Automated</p> <p>Add questions from CSV file containing question and reference answer columns.</p> <p>Click here for sample file.</p> <p>Post name</p> <input type="text"/> <p>Choose file No file chosen Add question</p>

Figure 21 Add questions

5.2.2.2 Posts

Questions are grouped together into posts so that they can be easier to view. Questions uploaded together will be added into the same post, either a new or an existing one.

ASAS QUESTIONS ANSWERS ADD QUESTIONS ADD ANSWERS POSTS MODEL STUDENTS LOGOUT			
ID	Name	Poster	Action
1	Manually added	Michelle Vanessa	Details
2	Basic	Michelle Vanessa	Details
3	Mohler v1.0	Michelle Vanessa	Details
4	Mohler v2.0	Michelle Vanessa	Details
5	Sample	Michelle Vanessa	Details

Figure 22 Admin posts list

ASAS QUESTIONS ANSWERS ADD QUESTIONS ADD ANSWERS POSTS MODEL STUDENTS LOGOUT			
Post name Mohler v2.0		Poster Michelle Vanessa	Back
ID	Question	Reference Answer	Action
44	What is the role of a prototype program in problem solving?	To simulate the behaviour of portions of the desired software product.	Details
45	What stages in the software life cycle are influenced by the testing stage?	The testing stage can influence both the coding stage (phase 5) and the solution refinement stage (phase 7)	Details
46	What are the main advantages associated with object-oriented programming?	Abstraction and reusability.	Details
47	Where do C++ programs begin to execute?	At the main function.	Details
48	What is a variable?	A location in memory that can store a value.	Details
49	Where are variables declared in a C++ program?	Variables can be declared anywhere in a program. They can be declared inside a function (local variables) or outside the functions (global variables)	Details
50	What is the main difference between a while and a do...while statement?	The block inside a do...while statement will execute at least once.	Details
51	What is typically included in a class definition?	Data members (attributes) and member functions.	Details
52	What is the difference between a data member and a local variable inside a member function?	Data members can be accessed from any member functions inside the class definition. Local variables can only be accessed inside the member function that defines them.	Details
53	What is the difference between a constructor and a function?	A constructor is called whenever an object is created	Details
54	When does C++ create a default constructor?	If no constructor is provided	Details
55	How many constructors can be created for a class?	Unlimited number.	Details

Figure 23 Admin post details

5.2.2.3 Answers

Answers from all students can be viewed in this page. Their scores can also be viewed from this page. However, scoring the answer can only be done in the questions list page.

ASAS QUESTIONS ANSWERS ADD QUESTIONS ADD ANSWERS POSTS MODEL STUDENTS LOGOUT						
ID	QuestionID	Answer	System score	Score 1	Score 2	
80	44	High risk problems are address in the prototype program to make sure that the program is feasible. A prototype may also be used to show a company that the software can be possibly programmed. 	-	3.5	3.5	
81	44	To simulate portions of the desired final product with a quick and easy program that does a small specific job. It is a way to help see what the problem is and how you may solve it in the final project.	-	5	5	
82	44	A prototype program simulates the behaviors of portions of the desired software product to allow for error checking.	-	4	4	
83	44	Defined in the Specification phase a prototype stimulates the behavior of portions of the desired software product. Meaning	-	5	5	
84	44	It is used to let the users have a first idea of the completed program and allow the clients to evaluate the program. This can generate much feedback including software specifications and project estimations of the total project.	-	3	3	
85	44	To find problem and errors in a program before it is finalized	-	2	2	
86	44	To address major issues in the creation of the program. There is no way to account for all possible bugs in the program	-	2.5	2.5	
87	44	you can break the whole program into prototype programs to simulate parts of the final program	-	5	5	
88	44	-To provide an example or model of how the finished program should perform. -Provides foresight of some of the challenges that would be encountered. - Provides opportunity To introduce changes To the finished program.	-	3.5	3.5	
89	44	Simulating the behavior of only a portion of the desired software product.	-	5	5	
90	44	A program that stimulates the behavior of portions of the desired software product.	-	5	5	
91	44	A program that simulates the behavior of portions of the desired software product.	-	5	5	
92	44	To lay out the basics and give you a starting point in the actual problem solving.	-	2	2	
93	44	To simulate problem solving for parts of the problem	-	4.5	4.5	
94	44	A prototype program provides a basic groundwork from which to further enhance and improve a solution to a problem.	-	2	2	

Figure 24 Admin answers list

Answers can also be added automatically by uploading a CSV file containing the answers, their questions, and the scores.

ASAS QUESTIONS ANSWERS ADD QUESTIONS ADD ANSWERS POSTS MODEL STUDENTS LOGOUT

Add answers

Add answers from CSV file containing question, answer, and score columns.

Click [here](#) for sample file.

No file chosen

Figure 25 Add answers

5.2.2.4 Neural network model

The neural network model is also attached to this system. The system will save the predicted scores by the model of the most recent training. When new data is added, the model can be trained using the new dataset by clicking the “Train model” button in this page. After training, all answers in the database will be predicted a new score using the new model. Lastly, the performance metrics will also be displayed in this page.

ASAS		QUESTIONS	ANSWERS	ADD QUESTIONS	ADD ANSWERS	POSTS	MODEL	STUDENTS	LOGOUT
Model details									
Evaluation method		Value							
Pearson		0.4639							
MAE		0.6124							
RMSE		0.9604							
		Train model							

Figure 26 Model details

5.2.2.5 Students

Admins can view all students in the system. From this page, the students can also be approved so that they can sign in using their accounts.

ASAS						QUESTIONS	ANSWERS	ADD QUESTIONS	ADD ANSWERS	POSTS	MODEL	STUDENTS	LOGOUT
ID	Full Name	Username	Email	Action		Approved							
13	Michelle Vanessa	vermichelleve	vermichelleve@gmail.com	Details	Approve								
14	First Last	lalaland	a@b.c	Details	Approve								
15	michelle vanessa	mich0069	mich0069@gmail.com	Details	Approve								
16	student 1	student1	student1@gmail.com	Details	Approve								
17	student 2	student2	student2@gmail.com	Details	Approve								
18	student 3	student3	student3@gmail.com	Details	Approve								
19	student 4	student4	student4@gmail.com	Details	Approve								
20	student 5	student5	student5@gmail.com	Details	Approve								
21	student 6	student6	student6@gmail.com	Details	Approve								
22	student 7	student7	student7@gmail.com	Details	Approve								
23	student 8	student8	student8@gmail.com	Details	Approve								
24	student 9	student9	student9@gmail.com	Details	Approve								
25	student 10	student10	student10@gmail.com	Details	Approve								
26	student 11	student11	student11@gmail.com	Details	Approve								

Figure 27 Students list

By clicking “Details” button, user can view all answers submitted by the student and their respective questions.

ASAS QUESTIONS ANSWERS ADD QUESTIONS ADD ANSWERS POSTS MODEL STUDENTS LOGOUT			
Name Auto Added		Email example@domain.com	
		Back	
ID	Question	Answer	Score
44	What is the role of a prototype program in problem solving?	To simulate the behaviour of portions of the desired software product.	-
45	What stages in the software life cycle are influenced by the testing stage?	The testing stage can influence both the coding stage (phase 5) and the solution refinement stage (phase 7)	-
25	What are the main advantages associated with object-oriented programming?	Abstraction and reusability.	-
46	What are the main advantages associated with object-oriented programming?	Abstraction and reusability.	-
26	Where do C++ programs begin to execute?	At the main function.	-
47	Where do C++ programs begin to execute?	At the main function.	-
27	What is a variable?	A location in memory that can store a value.	-
48	What is a variable?	A location in memory that can store a value.	-
67	How are arrays passed to functions?	by reference.	-
23	What is the role of a prototype program in problem solving?	To simulate the behaviour of portions of the desired software product.	-
24	What stages in the software life cycle are influenced by the testing stage?	The testing stage can influence both the coding stage (phase 5) and the solution refinement stage (phase 7)	-
28	Where are variables declared in a C++ program?	Variables can be declared anywhere in a program. They can be declared inside a function (local variables) or outside the functions (global variables)	-
49	Where are variables declared in a C++ program?	Variables can be declared anywhere in a program. They can be declared inside a function (local variables) or outside the functions (global variables)	-
29	What is the main difference between a while and a do...while statement?	The block inside a do...while statement will execute at least once.	-

Figure 28 Student details

5.2.3 Student

Student is another type of user in this system. A student can only view questions and submit answers to questions.

5.2.3.1 Questions

This page shows all questions available in the system. Student can only answer each question once, and the answer cannot be changed once submitted. The score of each answer is also displayed in this page.

ASAS QUESTIONS ACCOUNT POSTS LOGOUT				
ID	Question	Answer	Action	Score
23	What is the role of a prototype program in problem solving?	For testing purposes	Answer	
24	What stages in the software life cycle are influenced by the testing stage?	For testing purposes	Answer	
25	What are the main advantages associated with object-oriented programming?		Answer	
26	Where do C++ programs begin to execute?		Answer	
27	What is a variable?		Answer	
28	Where are variables declared in a C++ program?		Answer	
29	What is the main difference between a while and a do...while statement?		Answer	
30	What is typically included in a class definition?		Answer	
31	What is the difference between a data member and a local variable inside a member function?		Answer	
32	What is the difference between a constructor and a function?		Answer	
33	When does C++ create a default constructor?		Answer	
34	How many constructors can be created for a class?		Answer	
35	What is the difference between a function prototype and a function definition?		Answer	
36	What is the role of a header-file?		Answer	

Figure 29 Student questions list

ASAS QUESTIONS ACCOUNT POSTS LOGOUT

Question
What is the role of a prototype program in problem solving?

Answer
testing purposes

Submit Back

Figure 30 Student answer question page

5.2.3.2 Posts

Students can also view questions based on posts by clicking on the “Details” button on the rightmost column.

ASAS QUESTIONS ACCOUNT POSTS LOGOUT			
ID	Name	Poster	Action
1	Manually added	Michelle Vanessa	Details
2	Basic	Michelle Vanessa	Details
3	Mohler v1.0	Michelle Vanessa	Details
4	Mohler v2.0	Michelle Vanessa	Details
5	Sample	Michelle Vanessa	Details

Figure 31 Student posts list

5.2.3.3 Account

Users can view and update their account details from this page.

ASAS QUESTIONS ACCOUNT POSTS LOGOUT

Your account

First name	Last name
Michelle	Vanessa
Username	
vemichelleve	
Email	
vemichelleve@gmail.com	
Edit	Update password

Figure 32 Student account details

ASAS QUESTIONS ACCOUNT POSTS LOGOUT

Edit account

First name	Last name
<input type="text" value="Michelle"/>	<input type="text" value="Vanessa"/>
Username	
<input type="text" value="vemichelleve"/>	
Email	
<input type="text" value="vemichelleve@gmail.com"/>	
Save	Cancel

Figure 33 Student edit account

Chapter 6: Conclusion

This report discusses several approaches to improve the performance of the existing Siamese Bidirectional LSTM model in terms of grading accuracy of short-answered Data Structures questions.

Several techniques were implemented to the model, and based on the experiments conducted in this project, combination of batch normalization, L2 weight regularization, dropout, and question demoting yields the best result. When the techniques applied to the model, its performance is improved by 5.05% on Pearson correlation coefficient, 5.36% on MAE, and 2.14% on RMSE. The improvement can also be observed from the graphs of validation error against number of epochs (Figure 10 and Figure 14) as the improved model could perform predict with less error.

Furthermore, the model has also been converted to a multi-category classification model with 3 classes. As a result, the model has accuracy of 78.7% and F1 score of 74.35%, which are considerably low. The model does not perform as well as expected, so the regression model is still preferred in grading short answers.

Finally, a prototype system has been built to implement the improved model. The system provides a platform where users can upload questions, answer the questions, and grade the answers. Eventually, the submitted answers can be graded by the model or used as an additional training data. At the same time, the system could also be used as a platform for students to self-practice on Data Structures.

Reference

- [1] M. A. Sultan, C. Salazar and T. Sumner, “Fast and Easy Short Answer Grading with High Accuracy,” in *Proceedings of NAACL-HLT*, San Diego, California, 2016.
- [2] W. H. Gomaa and A. A. Fahmy, “Ans2vec: A Scoring System for Short Answers,” in *International Conference on Advanced Machine Learning Technologies and Applications*, 2019.
- [3] S. Kumar, S. Chakrabarti and S. Roy, “Earth Mover's Distance Pooling Over Siamese LSTMs for Automatic Short Answer Grading,” in *IJCAI*, 2017.
- [4] C. Leacock and M. Chodorow, “C-rater: Automated Scoring of Short Answer Questions,” *Computer and Humanities*, vol. 37, pp. 389-405, 2003.
- [5] M. Mohler and R. Mihalcea, “Text-to-text Semantic Similarity for Automatic Short Answer Grading,” in *Proceedings of the 12th Conference of the European Chapter of the ACL*, Athens, Greece, 2009.
- [6] F. S. Pribadi, T. B. Adj, A. E. Permanasari, A. Mulwinda and A. B. Utomo, “Automatic Short Answer Scoring Using Words Overlapping Methods,” in *Engineering International Conference*, 2016.
- [7] S. Hassan, A. A. Fahmy and M. El-Ramly, “Automatic Short Answer Scoring based on Paragraph Embeddings,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, pp. 397-402, 2018.
- [8] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.
- [9] S. Santurkar, D. Tsipras, A. Ilyas and A. Madry, “How Does Batch Normalization Help Optimization?,” 2018.
- [10] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” 2015.

- [11] G. E. Dahl, T. N. Sainath and G. E. Hinton, “Improving Deep Neural Networks for LVCSR Using Rectified Linear Units and Dropout,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.
- [12] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, “Improving Neural Networks by Preventing Co-adaptation of Feature Detectors,” 2012.
- [13] M. Mohler, R. Bunescu and R. Mihalcea, “Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, 2011.
- [14] C. Olah, “Understanding LSTM Networks,” 27 August 2015. [Online]. Available: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Accessed 10 March 2020].
- [15] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [16] Y. Goldberg and O. Levy, “word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method,” 2014.
- [17] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” *Advances in Neural Information Processing Systems 26*, 2013.
- [18] C. D. Manning, P. Raghavan and H. Schütze, “Tokenizer,” in *Introduction to Information Retrieval*, Cambridge University Press, 2008.
- [19] A. Casari and A. Zheng, *Feature Engineering for Machine Learning*, Sebastopol: O'Reilly Media, Inc., 2018.
- [20] Z. Huang, W. Xu and K. Yu, “Bidirectional LSTM-CRF Models for Sequence Tagging,” 2015.

- [21] D. Ho, E. Liang and R. Liaw, “1000x Faster Data Augmentation,” Berkeley Artificial Intelligence Research, 7 June 2019. [Online]. Available: https://bair.berkeley.edu/blog/2019/06/07/data_aug/. [Accessed 11 March 2020].
- [22] “Keras Documentation,” [Online]. Available: <https://keras.io/>. [Accessed 19 March 2020].
- [23] “pandas - Python Data Analysis Library,” [Online]. Available: <https://pandas.pydata.org/>. [Accessed 19 March 2020].
- [24] “NumPy,” [Online]. Available: <https://numpy.org/>. [Accessed 19 March 2020].
- [25] “scikit-learn: machine learning in Python,” [Online]. Available: <https://scikit-learn.org/stable/>. [Accessed 19 March 2020].
- [26] “Natural Language Toolkit,” [Online]. Available: <https://www.nltk.org/>. [Accessed 19 March 2020].
- [27] J. Perla, “Notes on AdaGrad,” 2014.
- [28] J. Bjorck, C. Gomes, B. Selam and K. Q. Weinberger, “Understanding Batch Normalization,” 2018.
- [29] “Regularization for Simplicity: L₂ Regularization,” Machine Learning Crash Course, 10 February 2020. [Online]. Available: <https://developers.google.com/machine-learning/crash-course/regularization-for-simplicity/12-regularization>. [Accessed 12 March 2020].
- [30] M. D. Shermis and J. Burstein, “Intellimetric TM: From Here to Validity,” in *Automated Essay Scoring: A Cross-Disciplinary Perspective*, Mahwah, New Jersey, Lawrence Erlbaum Associates, 2003, p. 71.
- [31] “Precision and Recall,” Precision and Recall, [Online]. Available: https://en.wikipedia.org/wiki/Precision_and_recall. [Accessed 20 March 2020].

- [32] “Django,” [Online]. Available: <https://www.djangoproject.com/>. [Accessed 14 March 2020].
- [33] “Django Documentation,” [Online]. Available: <https://docs.djangoproject.com/en/3.0/>. [Accessed 14 March 2020].
- [34] “What IS SQLite?,” SQLite, [Online]. Available: <https://www.sqlite.org/index.html>. [Accessed 22 March 2020].
- [35] “Django REST Framework,” [Online]. Available: <https://www.django-rest-framework.org/>. [Accessed 14 March 2020].

Appendix A: Experiment Results (Regression Model)

Remarks	Reg LSTM	Reg dense	Dense dropout	LSTM dropout	Batchno rm	LSTM batchnorm	Pearson	RMSE	MAE	Avg
Original code	0	0	0.5	0	Yes	No	0.4416	0.9815	0.6471	0.8143
Regularize LSTM	0.01	0	0.5	0	Yes	No	0.4512	0.9787	0.6293	0.8040
Regularize LSTM	0.001	0	0.5	0	Yes	No	0.4526	0.9556	0.6466	0.8011
Regularize LSTM	0.0001	0	0.5	0	Yes	No	0.4513	0.9563	0.6467	0.8015
Regularize perceptron	0	0.01	0.5	0	Yes	No	0.4471	0.9694	0.6309	0.8002
Regularize perceptron	0	0.001	0.5	0	Yes	No	0.4491	0.9677	0.6571	0.8124
Regularize perceptron	0	0.0001	0.5	0	Yes	No	0.4504	0.9952	0.6349	0.8151
Perceptron dropout	0	0	0.2	0	Yes	No	0.4436	1.0034	0.6369	0.8202
Perceptron dropout	0	0	0.6	0	Yes	No	0.4504	0.9778	0.6450	0.8114
LSTM dropout	0	0	0.5	0.2	Yes	No	0.4476	0.9697	0.6445	0.8071
LSTM dropout	0	0	0.5	0.4	Yes	No	0.4506	0.9632	0.6432	0.8032
LSTM dropout	0	0	0.5	0.5	Yes	No	0.4413	0.9599	0.6629	0.8114
Batch normalization	0	0	0.5	0	No	No	0.4142	0.9976	0.6582	0.8279
Recurrent batchnorm	0	0	0.5	0	Yes	Yes	0.4426	0.9648	0.6429	0.8039
Regularize	0.001	0.01	0.5	0	Yes	No	0.4447	0.9732	0.6481	0.8107
Dropout	0	0	0.6	0.4	Yes	No	0.4473	0.9845	0.6359	0.8102
Regularize & dropout	0.001	0.01	0.6	0.4	Yes	No	0.4486	1.0199	0.6294	0.8247

Regularize & dropout LSTM	0.001	0.01	0.5	0.4	Yes	No	0.4494	0.9504	0.6523	0.8014
Reg, dropout LSTM, batchnorm	0.001	0.01	0.5	0.4	Yes	Yes	0.4461	1.0000	0.6200	0.8100
Question demoting	0	0	0.5	0	Yes	No	0.469	0.97615	0.62045	0.7983
Question demoting, optimal	0.001	0.01	0.5	0.4	Yes	No	0.4639	0.9604	0.6124	0.7864
Question demoting, optimal & batchnorm	0.001	0.01	0.5	0.4	Yes	Yes	0.4601	0.9816	0.6091	0.7954