**SMART GRADING SYSTEM**

**Prachi Channe, Shruti Diwate, Dnyanada Mahajan, Sneha Kharate**

School of Computer Engineering and technology, MIT Academy of Engineering, Pune

**Abstract**

The understanding of writing skills of students is done by essay type of assignments given by teachers. Courses with hundreds of students, on the other hand, put a lot of pressure on professors because delivering individual grades is largely a manual, repetitive, and time-consuming task. This project aims to develop a system using the machine learning approach for suggesting computer-assisted grading in essays provided by teachers. When an essay is loaded into the proposed grading system, the system accepts the essay given as the input and grades it using deep learning techniques and its layers such as LSTM, Bi LSTM, Attention and dense layers. Proposed paper consists of a comparative study of models with different combination of LSTM, Bi LSTM and Attention layers.

**Keywords:** Machine learning,Neural Networks, LSTM, RNN, Grading, Bi LSTM, Attention Mechanism.

**1. Introduction**

Essay writing is usually a part of the student assessment process. Several organizations, such as Educational Testing Service (ETS)1, evaluate the writing skills of students in their examinations. Because of the large number of students participating in these exams, grading all essays is very time consuming. Thus, some organizations have been using AES systems to reduce the time and cost of scoring essays. Automated essay scoring refers to the process of grading student essays without human interference. An AES system takes as input an essay written for a given prompt, and then assigns a numeric score to the essay reflecting its quality, based on its content, grammar, and organization. Such AES systems are usually based on regression methods applied to a set of carefully designed features. The process of feature engineering is the most difficult part of building AES systems. Moreover, it is challenging for humans to consider all the factors that are involved in assigning a score to an essay.

This project aims at developing a model using machine learning techniques which automatically grades an essay upon submission. We grade our essay on a scale of 1-10 and the result would be displayed to the user. Our AES system, on the other hand, learns the features and relation between an essay and its score automatically. Since the system is based on recurrent neural networks, it can effectively encode the information required for essay evaluation and learn the complex patterns in the data through non-linear neural layers. Our system is among the first AES systems based on neural networks designed with a combination of different LSTM layers.

**2. Related Work**

This section describes the various approach to Automated Essay Grading using different type of Neural network Models. The approaches are explained in brief as follows:

The NN model for Automated Essay Grading [1] scored 0.9447875 during the Automatic Essay Grading Kaggle Competition (Nguyen, Neural Networks for Automated Essay Grading, 2016) on the quadratic weighted kappa metric. This neural network model employed a 300-dimensional Glove as the embedding layer initialization.

The study on essay scoring (Farah Nadeem, 2019) demonstrates how a neural network with cross-sentence dependencies and a discourse-based training objective can outperform both feature-based state-of-the-art models and hierarchical LSTMs in terms of automatic essay scoring for the LDC TOEFL essay data. The best results in (Liu et al., 2019) are achieved with a model that learns the combination of hand-crafted features and the neural document representation with Quadratic Weighted kappa of 0.852 and 0.736 for set 1 and set 2 respectively [2]

Research paper on essay scoring (A. Wiratmo and C. Fatichah 2020) presents novel LSTM dependency tree transfer learning scoring method for short essays in Indonesia. The LSTM [3]  architecture for essay grading can take both sequence and dependency into account. This proposed technique offers QWK and accuracy results of 53.68% and 16.23%, respectively.

The model (Zhu, Wilson & Sun, Yu 2020) developed a straightforward grading scheme that achieves a quadratic weighted kappa of 0.70026 using machine learning and natural language processing. The tokenized sequences are evaluated using an LSTM neural network, while the vector representation is evaluated using a 2-layer neural network [3]. Another 2-layer neural network is used to forecast the final score after the results are concatenated.

The machine learning model (Manvi Mahana in 2012) employed a model to train and test across each collection of essays. Within each essay set, a 5-fold cross validation was performed. The average kappa value for this was 0.73 [5].

**3. Smart Grading and Feedback System**

**3.1 Task description**

The objective of this research is to develop a machine learning-based, intelligent system that can grade essays on its own. A dataset with a significant number of essays on a certain topic should be carefully picked in order to ensure consistency among the raters. Our dataset has enough essays on various topics that have been graded. Pre-processing of the dataset is the next step. Cleaning the data is the first stage in the pre-processing process. The process of cleaning data involves removing any inaccurate, incomplete, duplicate, or other wrong data from the dataset. The removal of all characters from the dataset that aren't alphabets is the second step in the data cleaning process. The stop words are then all eliminated from the text. To get rid of stop words, the text is broken up into words, and those words are eliminated if they appear on the NLTK list of stop words.

Word tokenization is then applied to the words. The phrases, sentences, and paragraphs in this passage are divided into many units. Tokens are the name for these more compact objects. Then, these tokens are further examined in order to categorise or count them according to a specific sentiment. Then, using word embedding, we create a Word2Vec model in which the words or phrases are translated into real-number vectors. When words are embedded, those that share a semantically similar meaning are closer together than those that do not. These are than passed to the 3 layered LSTM model. There are total three models with different layer combination but common activation function and evaluation metrics. The activation metrics used is ReLU which is explained in the further section of this paper. We have used Quadratic Weighted Kappa as the Evaluation metric for the Models.

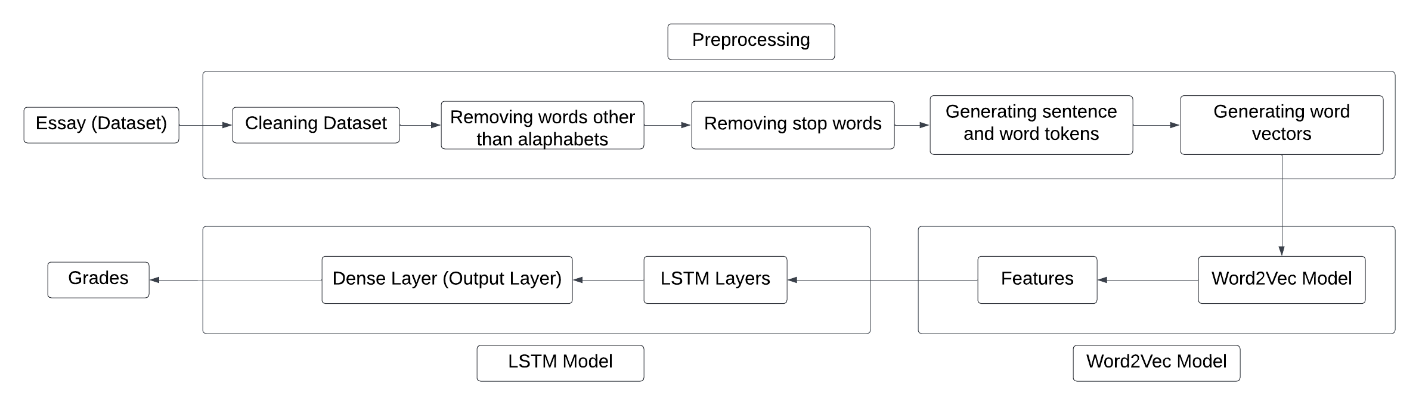


Figure 3.1: Data Flow

**3.2 Activation Function:**

Layers of nodes make up a neural network, which may be trained to map instances of inputs to outputs. The inputs are multiplied by the weights of a node and added together for a certain node. The node's total activation is known as this value. The activation total is then modified using an activation function, which determines the node's specific output or "activation.". The simplest activation function, where no transform is used at all, is known as the linear activation. A network that exclusively uses linear activation functions can be trained relatively quickly, but it is unable to learn complex mapping functions. It is preferable to use nonlinear activation functions since they enable the nodes to understand more intricate data structures. The sigmoid and hyperbolic tangent activation functions are two widely utilised nonlinear activation functions. The sigmoid and tanh functions both saturate, which is a general issue. This indicates that for tanh and sigmoid, high values snap to 1.0 and small values snap to -1 or 0. Additionally, sigmoid and tanh are only really sensitive to input changes around their midpoints, or 0.5 and 0.0 respectively. That’s why we decided to use ReLU activation function in our model.

ReLU is most commonly used activation function in machine learning models. Any negative input causes the function to return 0, but any positive value x causes it to return that value. Thus, it may be expressed as:

f(x)=max (0, x)

**3.3 Evaluation Metrics**

Quadratic Weighted Kappa

A set of predictions and a set of multiclass labels are measured by the Quadratic Weighted Kappa index. It attempts to take into account the similarity between the classes, beyond only the class, rather than just focusing on the precision of the match between predictions and labels. As a gauge of agreement between observed raters in cross-classification, Cohen's weighted kappa is frequently utilised. When ratings are given on nominal scales without an order structure, an appropriate index of agreement is used.Calculation of Cohen’s kappa may be performed according to the following formula:

K = (Pr(a)−Pr(e))/1−Pr(e)

Where Pr(a) represents the actual observed agreement, and Pr(e) represents chance agreement. Generally, a kappa of less than 0.4 is considered poor (a Kappa of 0 means there is no difference between the observers and chance alone). Kappa values of 0.4 to 0.75 are considered moderate to good and a kappa of >0.75 represents excellent agreement. A kappa of 1.0 means that there is perfect agreement between all raters.

**4. System Design**

This section introduces the proposed methodology, we have implemented Recurrent Neural Network (RNN) which are most commonly used neural network for problem solving by researchers. Basically, RNN is type of Neural Network which uses the output from previous layer to fed as input to the next layer. Paper consists of implementation and observation of the comparative working of Long Short-Term Memory (LSTM), Bidirectional-LSTM (Bi-LSTM) and Bi-LSTM with attention layer. LSTM is one of the types of RNN which is capable of solving complex problems. This section provides a brief description about the architecture and working of mentioned LSTM models and the training phase.

Word2Vec Model (Word embedding model) :

In Word2Vec model, each given essay consists a number of words, and each word represent by a word embedding according to word2vec. The embedding representations are expected to catch the semantic information carried by each word, i.e. the words with similar meanings will near to each other in the vector space.

For example, from our corpus, we can find that "computer" similar to "laptop".

The Neural Network can learn to identify the vector of the input word by using the surrounding word of this input. That means if two different words have the same context, the network tends to give them similar word vector. Hence, every word in our dataset has a unique vector containing the latent semantic and the vectors of the words in one essay can combine to an essay matrix which is the input unit of our scoring machine

**4.1 3 -Layer LSTM Model**

The paper, first introduces 4-layer LSTM model. It is a sequential model and works better for chosen dataset which is a corpus of essays in text format. The first layer of model is word embedding layer. The layer takes 300 as first argument which is number of features (output generated from word embedding layer i.e., Word2Vec model), dropout and recurrent dropout as 0.4 respectively and input size from 1 to 300 that is length of each sentence sequence. The next layer takes 64 features as first argument with recurrent dropout of 0.4. Third layer is also dropout layer with dropout value of 0.5. The final layer is dense layer, it reduces the dimensionality to 1 which is predicted score. Model have uses ReLU activation function in the dense layer so that the score can be predicted correctly since the values of ReLU function ranges form – ∞ to + ∞. For fitting of training data, model have been passed through batch size of 64 and 100 epochs. These epochs are varied according to the size of the test data. The model produces effective results. Diagrammatic representation is given in figure 4.1

**4.2 LSTM with Bi-LSTM**

To improve the performance / results got in Unidirectional LSTM, next we implemented the Bi-LSTM model. Unlike standard LSTM, here, the input is allowed to flows in both directions and it is capable of utilizing information from both sides. Again, it is powerful while modelling the sequential data, improving dependencies between words and sentences is both directions. It adds one more LSTM layer to previous model, the direction of information flow is revered. Then model combines the output of both LSTM layers to get the final output. Bi-LSTM is considered to produce more meaningful output when both LSTMs are combined. All other parameters are kept same. Experiments prove that Bi-LSTM model performs better than standard LSTM model. Diagrammatic representation is given in figure 4.2

**4.3 LSTM and Bi-LSTM with attention layer**

Though, using Bi-LSTM gave better results, in order to improve model performance, attention layer is introduced to the model. Sometimes, basic LSTM gets confused between the words and can predict wrong word. So, in order for the encoder to search for most relevant information, model have been introduced with additional layer of attention mechanism. By applying the attention mechanism, the model will be able to effectively extract the information between essays through inter-sentence alignment and gain better performance. Diagrammatic representation is given in figure 4.3

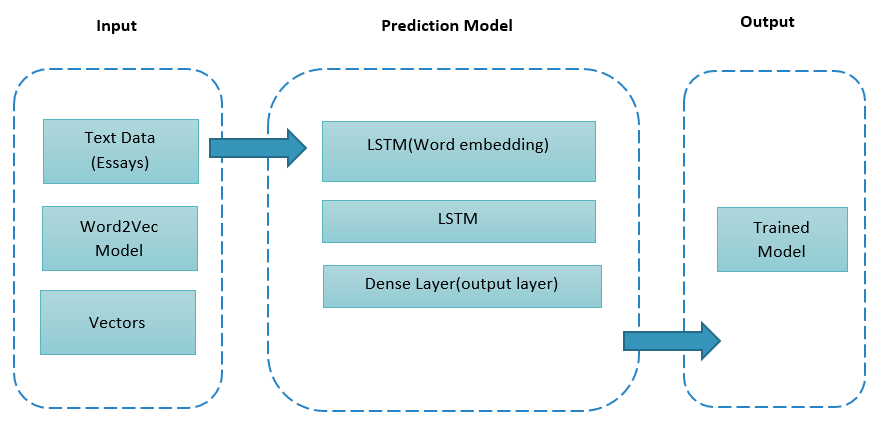
****

Figure 4.1: LSTM Model

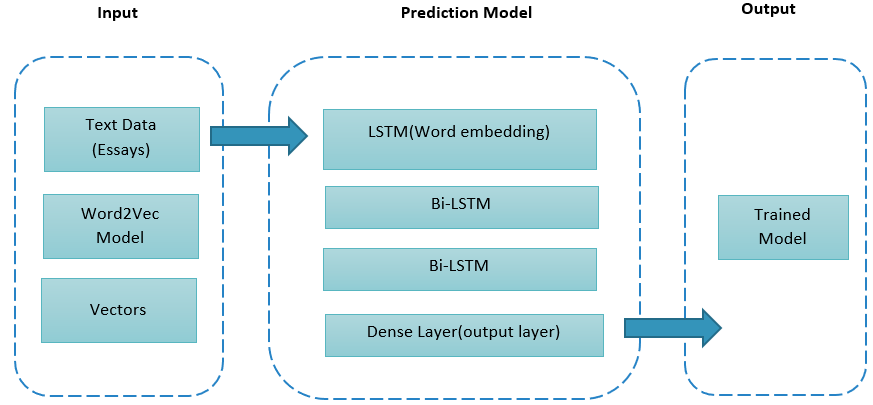


Figure 4.2: LSTM with Bi LSTM Model

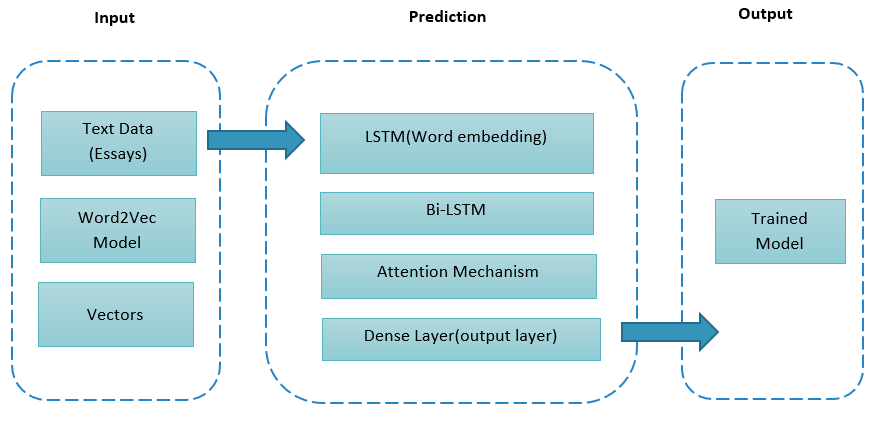
****

Figure 4.3: LSTM and Bi LSTM with Attention

**5. Experiment Setup**

Dataset used in this paper is “The Hewlett Foundation: Automated Essay Scoring” on Kaggle. The dataset includes 8 essay sets. Each of the essay sets was generated from individual prompt. Average length of essays is in range of 150 to 550 words per response. All essays were hand graded and were double-scored. The training data is in the format of tab-separated value (TSV) file. There are total 3 scores i.e., rater1 score rater2 score and domain score. The Domain score is addition of rater 1 and 2 scores. There are some unwanted empty columns also present in the dataset these columns were dropped using pandas’ libraries. Hence the final dataset has essay\_id, essay\_set, essay, domain1\_score columns.

After generation of dataset the data needed to be preprocessed before passing on to the model. The essays were first cleaned by removing the stop words, punctuation marks and converting all characters to lower case. Once the data is cleaned next step was to generate feature vector that is to be passed to the word2Vec Model. In order to generate the feature vectors first the cleaned essay was converted into sentence tokens and finally to word tokens. The output of Word2Vec model was passed as an input to the LSTM layers. For models other than neural network word embeddings were generated.

For selecting the best algorithm, various machine learning algorithms were used and the model with best kappa score is selected. following are the models used:

* Linear Regression
* Support Vector Regression (SVR)
* Long Short-Term Memory (LSTM)
* K-Nearest Neighbor
* Decision Tree Classifier

**6. Results and Discussions**

In this section the results of comparison of different machine learning models is explained and the comparison between different LSTM layer is also proposed. Figure 6.1 shows the results of different machine learning models out of which LSTM gives best performance in terms of QWK as well as MSE and variance. Hence the LSTM model was selected for further experiment purpose and predictions of grades.

|  |  |  |  |
| --- | --- | --- | --- |
| **Models** | **Kappa Score** | **Mean Squared Error** | **Variance** |
| Linear Regression | 0.85 | 20.94 | 0.73 |
| SVR | 0.64 | 38.19 | 0.53 |
| LSTM | **0.94** | 7.47 | **0.91** |
| K-Nearest Neighbours | 0.92 | 14.7 | 0.82 |
| Decision Tree Classifier | 0.86 | 22.56 | 0.71 |

Figure no. 6.1: Comparative Study of different machine learning models

Once the model was finalized, various combination of layers in the LSTM were used for better results. The models were modified with different LSTM, Bi LSTM and Attention layers. The combination of layers used are as follows:

* LSTM + LSTM
* LSTM + Bi LSTM + Bi LSTM
* LSTM + Bi LSTM + Attention

Hyperparameter for all of the models were set as constant with batch size of 64, 50 epochs and activation function as ReLU. 5-fold Cross validation was used in each model. The results of QWK, MSE and variance were noted. The results of this comparative study of layers are given in figure 6.2. It is observed that the model that contains the Attention layer performs best amongst all the three models. Further different activation function was used and the results were noted. The results are shown in figure 6.3. For sigmoid and tanh, the QWK is zero and the MSE is very large as compared to the other models. The noted results were not satisfactory and ReLU was decided to be used as activation function for building the final model.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Activation Function** | **Epochs** | **Batch size** | **QWK** | **MSE** | **Variance** |
| LSTM (2) | relu | 50 | 64 | 0.94 | 7.86 | 0.9 |
| LSTM + Bi LSTM (2) | relu | 50 | 64 | 0.95 | 6.66 | 0.92 |
| LSTM + Bi LSTM + Attention | relu | 50 | 64 | 0.96 | 6.2 | 0.92 |

Figure no. 6.2: Comparative Study of different LSTM Layers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Activation Function** | **Epochs** | **Batch size** | **QWK** | **MSE** | **Variance** |
| LSTM (2) | Sigmoid | 50 | 64 | 0 | 110.45 | 0 |
|  | tan h | 50 | 64 | 0 | 109.33 | 0 |
|  | relu | 50 | 64 | 0.94 | 7.86 | 0.9 |
|  |  |  |  |  |  |  |
| LSTM + Bi LSTM (2) | Sigmoid | 50 | 64 | 0 | 124.36 | 0 |
|  | tan h | 50 | 64 | 0 | 114.17 | 0 |
|  | relu | 50 | 64 | 0.95 | 6.66 | 0.92 |
|  |  |  |  |  |  |  |
| LSTM + Bi LSTM + Attention | Sigmoid | 50 | 64 | 0 | 114.68 | 0 |
|  | tan h | 50 | 64 | 0.95 | 112.23 | 0 |
|  | relu | 50 | 64 | 0.96 | 6.2 | 0.92 |

Figure 6.3: Using Different Activation function

Predictions were made using all the three models generated with selected random essay from the dataset. The results are shown in figure 6.5. It was observed that the model with LSTM, Bi LSTM and Attention layer performs well as compared to the other two models.

|  |  |  |  |
| --- | --- | --- | --- |
| **Essay Id** | **Model** | **Actual Score** | **Predicted Score** |
| 457 | LSTM (2) | 10 | 9 (9.16) |
|  | LSTM + Bi LSTM (2) | 10 | 9(8.558) |
|  | LSTM + Bi LSTM + Attention | 10 | 10(10.113) |

Figure 6.4: Predictions

QWK for prediction of all the selected model were calculated.

|  |  |
| --- | --- |
| **Model** | **QWK** |
| LSTM (2) | 0.924215844 |
| LSTM + Bi LSTM (2) | 0.956398147 |
| LSTM + Bi LSTM + Attention Mechanism | 0.962121614 |

Figure 6.4: Prediction model QWK

**6. Conclusion**

In this paper, we introduced a neural network model with a combination of different layers for essay scoring. For selection of the neural network model various machine learning algorithms were compared and the model with best kappa score was selected. After finalizing the model different combinations of LSTM layers were used to train the model. It was observed that the model with combination of all three layers i.e LSTM, Bi LSTM and attention gave good QWK and variance. Also, the predictions made by this model was best as compared to the predictions made by the other models. For selected best model for predictions amongst the three, we calculated the prediction QWK and it was observed that the Model with LSTM, Bi LSTM and attention Layer has the best QWK i.e 0.96.

**7. Future Scope**

Though our model gives good results with a Kappa Score of 0.96 there is always a scope to improve your model. Proposed Model works good with the typed essays, further it can be modified for handwritten essays as well as essays for essay in different languages. Not Only essays there are also others modes of question and answers for which this model can be modified for grading. Short answers, long answers specific to questions can also be graded.

**References**

1. Automated Essay Grading using Machine Learning Algorithm V.V.Ramalingam 1, APandian2. , Prateek Chetry3 and      Himanshu Nigam4 Department of Computer Science and Engineering, S.R.M. Institute of Science and  Technology, Chennai,  India - 603203
2. Jan Philip Bernius, Stephan Krusche, and Bernd Bruegge. 2021. A Machine Learning Approach for Suggesting Feedback in Textual Exercises in Large Courses. Proceedings of the Eighth ACM Conference on Learning @ Scale. Association for Computing Machinery, New York, NY, USA, 173–182. DOI:https://doi.org/10.1145/3430895.3460135
3. Urs Von Matt, “Kassandra:The Automatic Grading Ssytem”,Institute for Scientific Computing,ETH Zurich,Switzerland,1994
4. Z. Wang, J. Liu and R. Dong, "Intelligent Auto-grading System," 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), 2018, pp. 430-435, doi: 10.1109/CCIS.2018.8691244.
5. Neslihan Süzen, Alexander N. Gorban, Jeremy Levesley, Evgeny M. Mirkes, Automatic short answer grading and feedback using text mining methods, Procedia Computer Science, Volume 169, 2020, Pages 726-743, ISSN 1877-0509,
6. Aldea, A. I., Haller, S. M., & Luttikhuis, M. G. (2020). Towards grading automation of open questions using machine learning. In J. van der Veen, N. van Hattum-Janssen, H-M. Järvinen, T. de Laet, & I. ten Dam (Eds.), *Engaging, Engineering, Education: Book of Abstracts, SEFI 48th Annual Conference University of Twente (online), 20-24 September, 2020* (pp. 584-593). University of Twente.
7. R. Krithika and Jayasree Narayanan. 2015. Learning to Grade Short Answers using Machine Learning Techniques. In Proceedings of the Third International Symposium on Women in Computing and Informatics (WCI '15). Association for Computing Machinery, New York, NY, USA, 262–271. DOI
8. Dr. A. Mercy Rani 2021. Automated Explanatory Answer Evaluation Using Machine Learning Approach. *Design Engineering*. (Jun. 2021), 1181 - 1190.
9. Sanchez, Victor & Nerbonne, John & Verspoor, Marjolijn. (2013). Identifying Important Factors in Essay Grading Using Machine Learning..