Grading Descriptive Answer Scripts Using Deep Learning

Neethu George, Sijimol PJ, Surekha Mariam Varghese

Abstract*: *The evaluation of answer papers considering semantics is a complex process that requires great intellectual effort from evaluators. The lack of availability of expert evaluators makes the evaluation more time consuming. Nowadays, everything is automated. Hence, in order to reduce the effort during the evaluation of answer scripts an automated system is required to grade the answer scripts correctly. This paper presents a system for descriptive answer checking and grading application based on natural language processing and deep learning. Features are extracted to create a model from the human evaluated sample dataset of answer scripts. The proposed sequential model consists of LSTM-RNN layer which sequentially takes the glove vector representation in a sentence of each word and converts to embedding vector representation. Embedding vector corresponding to the glove vector of the last word will be the representation of the entire sentence in its semantic form. The sequential model consists of embedding layer, Long Short Term Memory layer, dropout layer and dense layer. The regularization technique, dropout reduces over fitting by preventing complex co-adaptations on training data. The softmax activation function in the dense layer (fully connected neural network layer) gives the one hot encoded score for each answer. The model can assign scores of non-evaluated descriptive answers by comparing it with answer key. This approach is very useful in valuation of essays, descriptive answer scripts, document similarity checking, and plagiarism detection.

Index Terms*: *Long Short Term Memory (LSTM), Machine Learning (ML), Recurrent Neural Network (RNN), Deep Descriptive Answer Scoring (D-DAS).

I. INTRODUCTION

One of the significant part of education is the examination which is a measure of students learning ability. After examination, the teachers spend most of their time for evaluating the marks of the students and the evaluation takes bulk usage of human effort, time and cost. An automated assessment evaluation system can reduce the efforts during the evaluation. Today, many automated evaluation systems exist and they analyze a piece of text based on semantics, spelling and context. The evaluation of descriptive answers is still an open problem. Major problem among the existing systems is their efficiency. The subjective nature of the answer scripts evaluation corresponds to variations in awarding of grades by different human evaluators, which is seen as an unfair method of grading by students. This difficulty of grading answer sheets can be rectified by answer script evaluation tools which grade answer scripts automatically. An automated assessment system must be capable of scoring the answer papers within the range of those awarded by human evaluators. It must be consistent in the way it grades the answer scripts and thus it can save the time and cost of evaluation. Currently there exists many automated essay evaluation systems based on keyword matching, sequence matching and using bag of words model. Nowadays, many researchers focus on the fastest and most accurate area of machine learning, i.e the deep learning. By introducing an automated assessment system using deep learning can improve the efficiency of answer evaluation. Here, a system is introduced to automatically score answer scripts. The main aim of this model is to extract the semantics to efficiently represent the text in answer scripts and develop a model from the key and evaluated answer scripts to grade non evaluated answer scripts using deep learning.

The objective of this model is to extract the semantics to efficiently represent the text in answer scripts and develop a model from the key and evaluated answer scripts to grade non evaluated answer scripts using deep neural networks. It is a combination of NLP and machine learning. The learning is done by using the Recurrent Neural Network and LSTM cells. Deep Neural networks are able to capture semantics of text in order to and the similarity between texts. For efficiently represent semantics of the sentences as embedding vectors we use LSTM-Recurrent Neural Networks .The embed-ding vector corresponding to the last word will be the entire representation of the sentence in its semantic form. The fully connected neural network layer automatically learns and predicts the scores for the semantic representation of the answer based on previous knowledge. Significance of the system is that it is useful for valuation of large number of answer scripts and fast valuation. It is a new step in the fields of information retrieval, document similarity, semantic evaluation and essay evaluation. The goal of the system is to replace the traditional human evaluation of the answer sheet that de-pends on several factor such as time, mindset, presentation style and so on.

Revised Manuscript Received on March 10, 2019.

Neethu George, Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam, Kerala, India.

Sijimol PJ, Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam, Kerala, India.

Surekha Mariam Varghese, Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam, Kerala, India.

II. RELATED WORKS

In 1996, by the request from American College Board, Ellis Page developed the first AES system, Project Essay Grader. The Project Essay Grader analyze the essays to determine a huge set of text features, e.g. to predict mark that human evaluators give, fourth root of essay length uses an approach with regression. It fails to detect the content related features of an essay by mainly aiming the surface structures and ignoring the semantic aspect of essays. In 1995, Ranjit Biswas [2] has explained an answer paper evaluation system based on fuzzy sets known as Fuzzy Evaluation Method(FEM). The paper compares the traditional evaluation approaches and automated grading approaches. A value marking using vector technique is used by Fem which uses a fuzzy technique of computer system. The Intelligent Assessment Technologies in the UK developed an automatic assessor called AutoMark [3]. AutoMark uses Information Extraction(IE) techniques to provide automatic marking. This system searches for specific content in the student's response. In 2003, Leacock and Chodorow [4] have implemented an automated short answer scoring system, C-rater. C-rater is used to score responses to content based short answer questions. It uses predicate argument structure, pronominal reference, morphological analysis and synonyms to assign full or partial credit to a short answer question. Bayesian Essay Test Scoring system (BETSY) is introduced by Rudner et al.in 2001 which can classify text based on trained material. This system determines the most appropriate classification using an enormous set of features. Then, the conditional probability of existence of each feature is predicated by the proportion of texts within each category that includes the feature [5]. Later in 2017, Philip E. Robinson et al.[6] have presented an online learning platform for assessment, teaching and learning of programming. Jamsheedh C. V [1] has implemented a basic answer script evaluation system by using NLP and machine learning tools. Here, features are extracted from human evaluated sample dataset of answer scripts and high weightage given answer key. Panchami K. S [7] has implemented a diagram evaluation system using Support Vector Machines. The system uses Machine learning, Image processing and Natural language processing to effectively evaluate diagrams present in answer scripts. Charusheela Nehete et al. [8] followed another kind of research line that led to the Checkpoint, a NLP based descriptive answer checking and grading application. The online assessment system stores the answers as text files after the assessment for processing. It has large execution time and not able to handle very complex sentence structures. By examining spelling correction system in real world, Alexander et al. [9] com-pare five different measures semantic distance or similarity. Pantulkar S et al. [10] have explained a paper that computes the similarity between sentences based on wordnet. They show that measuring semantic similarity of sentences is closely related to semantic similarity between words. Ming Che Lee et al. [11] have presented a paper that contains grammar and semantic corpus based similarity algorithm for natural language sentences. In order to surmount the problems addressed, sentence similarity algorithm takes advantage of corpus-based ontology and grammatical rules. Atish and Vijay [12] have presented a paper based on calculation of semantic similarity between two words, paragraphs or sentences. For comparison the algorithm at first disambiguates all the sentences to make sure the correct meaning of the word. By the use of lexical database, an edge based approach is then introduced to calculate the semantic similarity between words and sentences [13]. Similar to the proposed D-DAS model for sentence embedding, the DSSM [14] and CLSM [15] models are developed for information retrieval. However, DSSM treats the input sentence as a bag-of-words and does not consider word dependencies. CLSM treats a sentence as a bag of n-grams, where n is defined by a window, and can capture local word dependencies. LSTM networks were developed in [16] to address the difficulty of capturing long term memory in Recurrent Neural Network. LSTM has been successfully applied to speech recognition [17, 18], which attains state of art performance. Hamid Palangi et al. [19] were presented a sentence embedding method using LSTM Cells and Recurrent Neural Networks for fast information retrieval. It takes one-hot vector representation of the words of the sentence as input word vectors sequentially to extract its information, and to embed it into a semantic vector. When final word is reached, the semantic representation of the whole sentence is given by the hidden layer of the network. Sutskever et al. [20] have proposed English to French converter using neural networks. Here a LSTM-RNN converts input English sentence to a vector representation and another LSTM-RNN generates an output French sentence. Nitish et al. [21] have been introduced a regularization technique using dropout to overcome overfitting. During training the neural network may contain several unused units and the key idea of dropout is to randomly drop these units along with the connections in between these units. This prevents units from coadapting. At the time of training, dropout samples from vector is given as input to the trained model based on exponential number of different "thinned" networks. During test time, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single unthinned network that has smaller weights.

III. PROPOSED WORK

Many schemes and methods are currently available for evaluation of essays. But automatic evaluation and grading is not attempted successfully for descriptive answer scripts. Many concepts of NLP have been applied in finding the relevant words, extracting the meaning and assign a grade for different answers. D-DAS model (Deep Descriptive Answer Scoring model), predicts scores for the descriptive answers. The system takes the entire short answer as input and converts it into glove vector representation by using embedding layer. The LSTM RNN will learn the temporal data from the embedding layer and the embedding vector corresponding to the final glove vector will be the semantic representation of the entire answer. This is given as input to the dropout layer and then to the fully connected neural network layer with a softmax activation function.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Final layer will then predict the score. The proposed Deep Descriptive Answer Scoring model (D-DAS model) is a sequential model that consists of consists of embedding layer, LSTM-RNN layer, dropout layer and dense layer. The dense layer gives the one hot encoded score for each answer. The stages in the proposed model is as depicted in Fig. 1.

The developed system is comprised of three stages- Preprocessing, Semantic Extraction, Classification and Grading. The proposed dataset is taken as the input for the training phase. The preproc- essing phase converts the answers in the dataset into a set of index to glove vectors corresponding to each sentence. The Semantic Extraction phase makes the semantic representation of the entire answer by using the Embedding layer and LSTM-RNN layer.

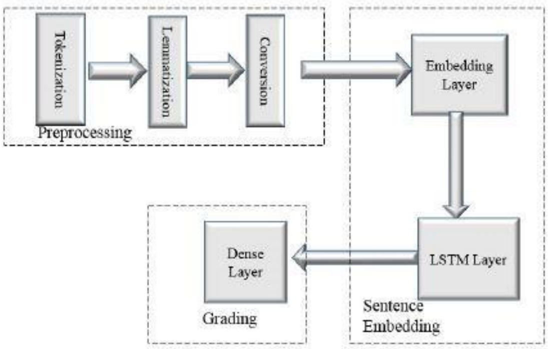

Fig. 1. Design of various modules in D-DAS model.

The embedding vector representation from the LSTM-RNN model is further used for the classification and grading phase. The testing deals with the scoring of an answer dataset based on the learned data in the trained model. The same processing has been applied to obtain the semantic representation of the answer and the embedded vector is given as input to the trained model. Based on the acquired knowledge the sequential model predicts the score for each answer. The basic architecture of the proposed system is shown in Fig. 2.

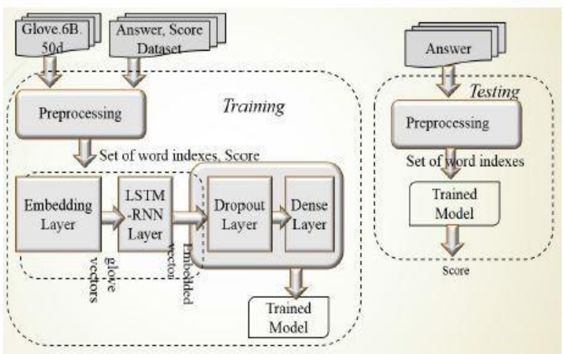

Fig. 2. System Architecture of D-DAS Model.

Different steps in the proposed model is explained below:

A. Preprocessing

Preprocessing stage extracts the relevant features of the text in the answer scripts and converts them into glove vector indexes. A dataset is prepared from the key and answer scripts with its corresponding human evaluated score. Glove.6B contains pre-trained word vectors of Wikipedia 2014 and Gigaword 5 that are used for obtaining the vector representation of the words in the sentences. The features of the answer are extracted after tokenization and lemmatization. The scores can be converted to any of the vector representation such as one hot vector.

B. Sentence Embedding

The sentence embedding module converts the set of glove indexes into its semantic representation, the embedding vector. The embedding layer simply converts the glove index corresponding to a word into its glove vectors. Let an answer contains n words then n is the sequence length. So n LSTM-cells will be there. Each glove vector is a 50/100/200 dimensional vector. The embedding vector representing the last word, x(n) in answer will be the semantic representation of the entire answer.

The main idea of using RNN for sentence embedding is to find a dense and low dimensional semantic representation by sequentially and recurrently processing each word in a sentence and mapping it into a low dimensional vector. The global contextual features of the whole text will be in the semantic representation of the last word in the text sequence.

The architecture of LSTM illustrated in Fig. 3 is used as sentence embedding method for the proposed system. In this figure, i(t), o(t), c(t) and f(t) are input gate, output gate, cell state vector and forget gate respectively. Wp1, Wp2 and Wp3 are peephole connections. Wi, Wreci and bi for i = 1, 2, 3, 4 are input connections, recurrent connections and bias values, respectively. s(.) and h(.) are tanh(.) function and (:) is the sigmoid function. The architecture is used to find y for each word, then use the y(n) representing the last word in the sentence as the n vector representing semantics for the whole sentence. Architecture of the proposed system is shown in Fig. 2.

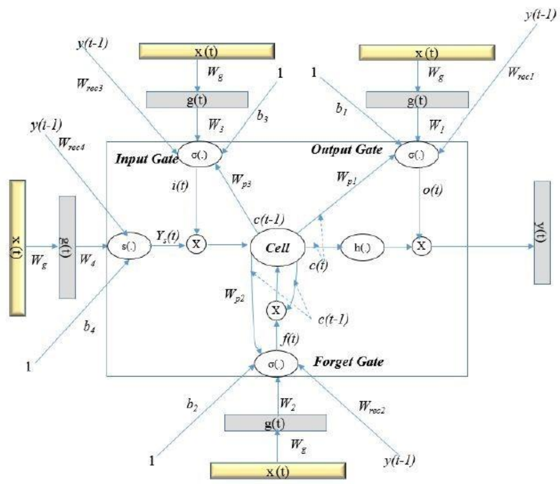

Fig. 3. Sentence Embedding using LSTM-RNN.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

C. Grading

The embedding vector from the LSTM layer will be the semantic representation of the answer. Based on this value the output layer, fully connected neural network layer, the dense layer will predict the one- hotted score. Supervised training is used for this sequential model. The neural networks are called from Keras library.

D. Dataset Preparation

The D-DAS dataset is the input to the preprocessing module. The dataset is loaded as a csv file. The dataset format is [question ID], [keyID], [answer], [mark]. Each row in the answer and mark columns of dataset file refers to a short answer and its evaluated mark respectively. Dataset is created for two different questions. For each question, around 50 unique answers are collected from different students to create the dataset. A sample dataset used in the D-DAS model is shown in Fig. 4.

	qid	keyid	answerId	answer	mark
0	1	1	1	timeliness:when to standardize language.confor...	3.0
1	1	1	2	The three issues are: timeliness conformance a...	3.0
2	1	1	3	timeliness:when to standardize language.	1.0
3	1	1	4	timeliness:when to standardize language.confor...	2.0
4	1	1	5	timeliness:when to standardize language.obsole...	2.0

Fig. 4. Dataset.

IV. ALGORITHM

The Algorithms 1,2and 3 describes the entire steps followed in the model. The procedure CreateWord To ID Dictionary is used to create the Word To ID dictionary for index lookup. The Algorithm 3, converts the answer and mark dataset to word id and one-hotted score respectively. The sequential model is created constitutes 4 layers- embedding layer, lstm layer, dropout layer and dense layer. The model is trained by using the Xtrain and Ytrain values. During testing Xtest and Ytest values are used. Xtrain and Xtest values are taken in the ration 80:20 from the dataset.

Algorithm 1 : Training and Testing the D-DAS Model

1: Call Create Word To ID Dictionary
2: Call Preprocess Data
3: Read the Input Data and Output Data
4: Split input Data to Xtrain, Xtest and output Data to Ytrain, Ytest
5: Read GloVe.6B.50d dataset
6: Remove first column and Convert it into GloVe matrix
7: First column represents the word and the remaining 50 columns represent the 50d vector
8: Create the sequential Model with 4 layers- embedding layer, lstm layer, dropout layer and dense layer.
9: Pass the dimensions of the GloVe matrix, sequence length, Xtrain, Ytrain to embedding layer
10: Make use of the trained model to predict the scores for the Xtest values

Algorithm 2 : Creating Word_To_ID_Dictionary

1: **procedure** CREATEWORD_TO_ID_DICTIONARY
2: Read GloVe.6B.50d dataset
3: . GloVe.6B.50d dataset of the form 399999(words) X 51(columns)
4: Convert it into of the form word, vectors
5: create dictionary Word_To_ID
6: *Word_ To_ ID* ⟵ (*word; row index*)
7: **end procedure**

Algorithm 3 : Preprocessing Input and Output Data

1: **procedure** PREPROCESSDATA
2: Read Answer, mark to AnsDataset
3: **for** Ai in AnsDataset **do**
4: Tokenize Ai
5: Lemmatize Ai
6: **for** *wordj* 2inAi **do**
7: Find Id for *wordj* from Word To ID Dictionary
8: Add Id to *wordId* and words that not found to *wrongWords*
9: **end for**
10: **end for**
11: Set the sequenceLength . to get an optimum value for the input
12: Pad Sequences if maximum Length ofanswer<= sequence Length
13: Convert mark to one-hot encoding format
14: Save wordId as inputData, one-hotted Score as Output Data
15: **end procedure**

V. RESULT ANALYSIS

The proposed D-DAS model is a prediction based model which use neural networks to predict the output. 50 iterations and 100 epochs were used for training and experimented on 500 epochs. Table 1. shows the accuracy obtained for each epochs for the different LSTM-RNN types in D-DAS model. For 90% training data, the accuracy obtained by the D-DAS model using simple LSTM is 83%, using deep LSTM is 82% and bidirectional LSTM is 89%. The results show that the training dataset depends on their prediction accuracy.

The accuracy is calculated by using the accuracy score function in the metric module of scikit-learn. The accuracy score function computes the accuracy, either as the fraction or the count of predictions that are found to be correct. If the whole set of predicted labels for a sample strictly match with the true set of labels in multilabel classification, the subset accuracy is 1.0, otherwise it is 0.0.

Table 1. Accuracy of various model implementations

Training Percentage	Simple LSTM	Deep LSTM	Bi-directional LSTM
16	30	26	29
30	44	44	48
40	71	60	64
50	74	68	73
60	68	70	77
70	80	80	86
80	83	81	88
90	83	82	89

Fig. 5 shows the accuracy obtained for each epochs for the different LSTM-RNN types. A simple LSTM contains only one LSTM layer whereas in a deep LSTM, many LSTM layers will be there for sentence embedding. LSTM in its core, preserves information from inputs that has already passed through it using the hidden state. Unidirectional LSTM only preserves information of the past because the only inputs it has seen are from the past. Cohen's kappa is used to measures inter- annotator agreement. It is defined as

$$K = (Po - Pe)/ (1= Pe) \qquad (1)$$

where Po is the empirical probability of agreement and Pe is the expected agreement when both annotators assign labels randomly. Pe is estimated using a per-annotator empirical prior over the class labels. The kappa statistic, which is a number between 1 and 1. The maximum value

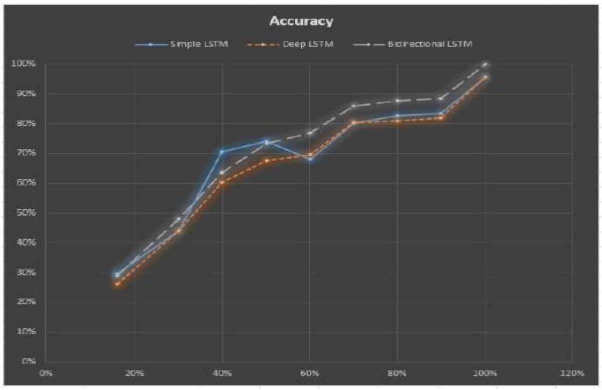

Fig. 5. Training Percentage v/s Accuracy.

means chance agreement. This measure is intended to compare labelings by different human annotators, not a classifier versus a ground truth. The kappa score is a number between -1 and 1. Scores above 0.8 are generally considered good agreement, zero or lower means no agreement. Kappa scores can be computed for binary or multiclass problems, but not for multi label and not for more than two annotators. Table 2. contains the kappa scores for different LSTM-RNN models.

Table 2. Kappa score of various D-DAS Model implementations.

Training Percentage	Simple LSTM	Deep LSTM	Bi-directional LSTM
16	-0.0341	-0.02	0.135
30	0.4645	0.27	0.323
40	0.6743	0.4576	0.6546
50	0.672	0.602	0.7277
60	0.7767	0.714	0.88
70	0.786	0.72998	0.91415
80	0.84	0.82	0.91387
90	0.866	0.8817	0.9333

A comparative study of kappa scores for the different proposed neural network models are depicted in Fig. 6. The results indicates that the D-DAS model with Bi-LSTM has higher performance than the other neural network based models. It reaches to the good agreement value for 60% training data.

The performance evaluation results prove that neural networks are suitable for doing predictions from huge datasets. An average of 73% accuracy was obtained for the developed D-DAS model with Bi-LSTM layer.

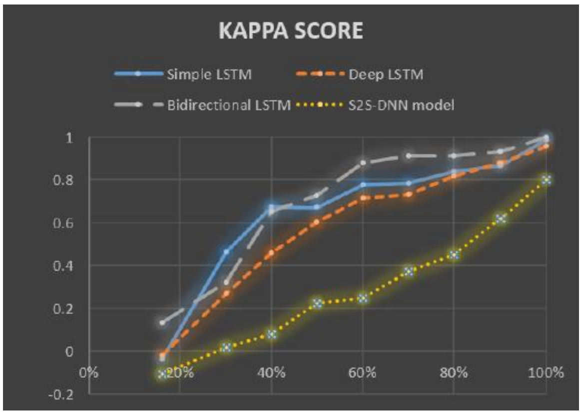

Fig. 6. Kappa score of different proposed models

VI. CONCLUSION

The answer paper evaluation is a tedious and time consuming process for many evaluators. For many teachers to finish the evaluation of all student answers in a short interval time is the major challenging task. Currently answer script evaluation is one of the on-going research area. The proposed system is an automated descriptive answer checking and grading application using deep learning. For semantics interpretation and grading of descriptive answers, natural language processing and deep learning tools have been used.

The dataset for the evaluation was prepared manually from fifty distinct answer scripts. Features are extracted to create a model from human evaluated sample dataset of answer scripts and high score given key. A model known as Deep Descriptive Answer Scoring Model is performed which outperforms all other models. The Deep Descriptive Answer Scoring Model consists of LSTM- Recurrent Neural Networks that sequentially takes the glove vectors of each word in the answer and converts it into the semantic representation. Hence, the embedding vector corresponding to the last word will be the semantic representation of the entire answer. The accuracy obtained in each model has been compared with that of the existing system. From the findings, it is evident that the proposed models works better than the existing system.

REFERENCES

1. Jamsheedh C. V, Aby Abahai T and Surekha Mariam Varghese, “A Fair Assessment System For Evaluation And Grading Of Text In Degitized Descriptive Answer Scripts,” RACCCS, vol. 4, pp. 346- 352, 2016.
2. R. Biswas, “An application of fuzzy sets in students’ evaluation,” ELSEVIER Fuzzy Sets and Systems, vol. 22, pp. 229- 236, 1995.
3. P. B. Tom Mitchell, Terry Russell and N. Aldridge, “Towards robust computerised market-ing of free-text responses,” In Proceedings of the 6th CAA Conference Loughborough 2002.
4. C. Leacock and M. Chodorow, “C-rater: Automated scoring of short answer questions,” Computers and the Humanities, p. 389405, 2003.
5. L. M. Rudner and T. Liang, “Automated essay scoring using bayes’ theorem,” The Journal of Technology, Learning, and Assessment, vol.1, june 2002.
6. P. E. Robinson and J. Carroll, “An Online Learning Platform for Teaching,Learning,and Assessment of Programming,” IEEE Global Engineering Education Conference (EDUCON), vol. 3, pp. 547- 556, 2017.
7. Panchami K.S, Surekha Mariam Varghese and Aby Abhahai T, “Grading of Diagrams in Answer Scripts Using Support Vector Machine,” International Journal of Control Theory and Applications, vol. 10, pp. 345-350, 2017.
8. S. U. Charusheela Nehete, Vasant Powar and J. Wadhwani, ”Checkpoint An Online De-scriptive Answers Grading Tool,” IJARCS, vol. 8, pp. 135- 155, 2017.
9. Budanitsky and G. Hirst, ”Semantic distance in wordnet:an experimental, application-ori-ented evaluation of five measures,” Proc. Workshop WordNet and Other Lexical Resources, Second Meeting North Am. Chapter Assoc. for Computational Linguistics, June 2001.
10. P. Sravanthi and D. B. Srinivasu, ”Semantic Similarity Between Sentences,” International Research Journal of Engineering and Technology (IRJET), vol. 2, pp. 156- 161, 2017.
11. J. C. Ming Che Lee and T. C. Hsieh, ”A Grammar-Based Semantic Similarity Algorithm for Natural Language Sentences,” Hindawi Publishing Corporation Scientific World Journal, vol. 2014, pp. 1- 1, 2014.
12. A. Pawar and V. Mago, ”Calculating the similarity between words and sentences using a lexical database and corpus statistics,” IEEE Transactions on Knowledge and data Engineer-ing, February 2018.
13. Z. A. B. J. D. O. Yuhua Li, David McLean and K. Crockett, ”Sentence similarity based on semantic nets and corpus statistics,” IEEE Transactions on Knowledge and data Engineer-ing, vol. 18, pp. 1138- 1150, August 2006.
14. J. G. L. D. A. A. P.S. Huang, X. He and L. Heck, ”Learning deep structured semantic models for web search using clickthrough data,” in Proceedings of the 22nd ACM International Conference on Information Knowledge Management, pp. 2333-2338, 2013.
15. J. G. L. D. Y. Shen, X. He and G. Mesnil, ”A latent semantic model with convolutional-pooling structure for information retrieval,” CIKM ’14 Proceedings of the 23rd ACM Inter-national Conference on Conference on Information and Knowledge Management, pp. 101- 110, November 2014.
16. S. Hochreitur and J. Schmidhuber, ”Long short-term memory,” Neural Computation, vol. 9, pp. 1735- 1780, November 1997.
17. F. B. Hasim Sak, Andrew Senior, ”Long short-term memory recurrent neural network archi-tectures for large scale acoustic modeling,” in Proceedings of the Annual Conference of International Speech Communication Association, 2014.
18. A.. M. Alex Graves and G. Hinton, ”Speech recognition with deep recurrent neural net-works,” in Proc. ICASSP, Vancouver, Canada, May 2013.
19. Y. S. J. G. X. H. J. C. X. S. Hamid Palangi, Li Deng and R. Ward, ”Deep Sentence Embed-ding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 2, pp. 2329- 290, 2015.
20. O. V. I. Sutskever and Q. V. Le, ”Sequence to sequence learning with neural networks,” In Proceedings of Advances in Neural Information Processing Systems, pp. 3104- 3112, 2014.
21. A.K. I. S. Nitish Srivastava, Geoffrey Hinton and R. Salakhutdinov,” Dropout: A simple way to prevent neural networks from overfitting,” Journal of Machine Learning Research, pp. 929- 1958, 2014.

AUTHORS PROFILE

Neethu George received Bachelor of Technology in Computer Science and Engineering from Government Engineering College Idukki in 2017 and currently pursuing Master of Technology in Computer Science and Engineering from Mar Athanasius College of Engineering, Kothamangalam affiliated to APJ Abdul Kalam Technological University. Her research interest is in Deep Learning and Data Mining.

Dr. Surekha Mariam Varghese is currently heading the Department of Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam, Kerala, India. She received her B-Tech Degree in Computer Science and Engineering in 1990 from College of Engineering, Trivandrum affiliated to Kerala University and M-Tech in Computer and Information Sciences from Cochin University of Science and Technology, Kochi in 1996. She obtained Ph.D in Computer Security from Cochin University of Science and Technology, Kochi in 2009. She has around 27 years of teaching and research experience in various institutions in India. Her research interests include Machine learning, Network Security, Database Management, Data Structures and Algorithms, Operating Systems and Distributed Computing.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication