JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 1108 - 1113
https://publishoa.com
ISSN: 1309-3452

Automated Subjective Answer Evaluation System

Prof. Sharayu Lokhande
Computer Engineering Army Institute Of Technology
Pune, Maharashtra slokhande@aitpune.edu.in

Udit Chaudhary
Computer Engineering Army Institute Of Technology
Pune, Maharashtra uditchaudhary 18203@aitpune.edu.in

Akash Singh
Computer Engineering Army Institute Of Technology
Pune, Maharashtra aakashsingh 18136@aitpune.edu.in

Pranay Gaikwad
Computer Engineering Army Institute of Technology
Pune, Maharashtra
pranaygaikwad_18171@aitpune.edu.in

Himanshu Guleria
Computer Engineering Army Institute of Technology
Pune, Maharashtra himanshuguleria_182111@aitpune.edu.in

Prof Shilpa Pawar
EnT Engineering
Army Institute of Technology, Pune, Maharashtra
spawar@aitpune.edu.in

ABSTRACT

In this paper, we have studied LSTM (Long Short- Term Memory) network and presented a siamese adaptation of it for labelled data composed of variable-length pattern and pairs. Our model first takes in right answer and then assesses semantic similarity between the right answer and the given answer. In order to accomplish these we use word embedding vectors which are supplemented with synonymic information to the LSTMs. These vectors encode the expressed underlying meaning of the sentence which is of fixed size. The wording and syntax are also taken care of. We limit subsequent operations that rely on the simple Manhattan metric. The model's learned sentence representations are compelled to a highly structured space. The geometry of this space represents complex semantic relationships. Our results show that LSTM's can be really powerful language models and are especially suited to tasks which require intricate understanding.

Index Terms—RNN, LSTM, NLP

I. INTRODUCTION

Examining and evaluating answer sheets are time-consuming testing tools for assessing academic achievement, integration of ideas, and recall; however, manually generating questions and evaluating responses is costly, resource-intensive, and time-consuming. Manual evaluation of answer sheets takes up a notable number of instructors, a lot of valuable time and so it is a high-cost task. Also, different security concerns regarding paper leakage is one of the other challenges to conquer. The goal of this project is to create an automated examination system using machine learning, the natural language toolkit (NLTK), and the Python environment, Recurrent Neural Networks and web technologies to provide an inexpensive alternative to the current examination system. Understanding and retrieving information from text are crucial activities that can be considerably aided by modeling the underlying semantic similarity of sentences. Different wording/syntax used to convey the same meaning should not readily influence a successful model. The study of such a semantic word-based similarity criterion has consequently spawned a plethora of studies (Marelli et al. 2014). This has always been a challenging endeavour due to a lack of labelled data, sentences of different length and complexity, and bag-of-words/tf-IDF models, which are popular in natural language processing (NLP) but finite in this context due to their term-specificity (cf. Mihalcea, Corley, and Strapparava 2006). Because they are well-suited to variable-length inputs such as sentences, recurrent neural networks (RNN), particularly Hochreiter and Schmidhuber's (1997) Long Short-Term Memory model, have been particularly successful in this setting for language translation (Sutskever, Vinyals, and Le 2014) and tasks such as text classification (Graves 2012). Practically, by utilizing memory cell components that can store and access information across very long input sentences, the LSTM outperforms basic RNNs for learning long span dependencies. The LSTM, like RNNs, sequentially updates a hidden-state representation, but these events are also dependent on a memory cell that contains four additional components (all of those are real-valued vectors): a memory state cell, an output port of that controls how the memory state influences the other units, and an input (and forget) gate it (and ft) that defines what is cached in (and excluded from) memory based on each updated input and the current state.

II. RELATED WORK

Claudia Leacock and Martin Chodorow presented a paper titled “C-rater: Automated scoring of short answer questions”. They presented the idea of an automated scoring engine which scores the content based responses. It takes into account morphology, semantics, concept matching, typos and spelling correction etc and grades accordingly. The scores given by C-rater agreed with human grades 84% of the time. Ms. Sharmeen J. Shaikh, Ms. Prerana S. Patil presented a paper titled “Automated descriptive answer evaluation system using machine learning” . Their idea was to compare answers by calculating the comparing the weightage of keywords. A model answer was required and other answers were evaluated by comparing them to model answer. It also took into account the grammatical errors and linguistics.

Neethu George, Sijimol PJ, Surekha Mariam Varghese presented a paper titled “Grading Descriptive answer scripts using deep learning”. Their method also uses an answer key or a sample set of having correct answers and all the other answers are compared to these dataset of model answers. This is achieved using natural language processing and deep learning. A model is created using the dataset by extracting features. This model is used to assign scores by comparing it to answer key.

Piyush Patil, Sachin Patil presented another technique for subjective answer evaluation. Their algorithm performed tokenisation, chunking, clinking, lemmatization and word- netting, speech tagging to evaluate the text. A semantic meaning of context was also generated by the algorithm. Nave bayes classifier was used in this context.

Deep learning techniques can discover hidden compositions and features from training data at various levels of abstractions, which is important for comparing the text by exploiting deep architectures. Hinton and Salakhutdinov used a deep network to expand the LSA model to identify the query's and the document's hierarchical semantic structure.

Deep convolutional neural networks (CNN) have been used successfully to process voice, natural language, and images. This is the first successful attempt at using convolutional neural network-like algorithms to an information retrieval model.

By assuming that a question and it’s document have the same document-topic grouping a Bi-Lingual Topic Model (generative model) is put forward for Web search in The Bi-Lingual Topic Model gives superior performance over Probabilistic Latent Semantic Analysis by learning the model having questions-title pairs.

Huge corpus is a method for calculating sentence similarity that is based on analytic data of words. It is one of the more recent agile topics of research that has contributed to sentence similarity computation. The Hyperspace Analogues to Language (HAL) model and latent semantic analysis (LSA) are two common corpus-based similarity techniques.

The descriptive features-based method is another method in sentence similarity computations. The goal of the feature vector technique is to express a sentence with a limited number of attributes. Feature vector techniques differ in how they establish essential features and composite features.

Hyperspace Analogues to Language is another important corpus-based technique (HAL). HAL and LSA are similar in that they both use lexical co-occurrence information to understand the meaning of a word/text. In contrast to LSA, which generates an information matrix composed of words by text units of documents or paragraphs, HAL creates a word-by-word matrix based on word co-occurrences inside a preset length moving frame. The window (with a length of 10 words) advances the collection's whole text. A N x N matrix is generated for a vocabulary of N words. As the collection progressed, all of the matrix's entries recorded the (weighted) word co-occurrences within the window. The meaning of a word is then represented as a 2N-dimensional vector by combining the relevant row and column in the matrix. Finally, all of the phrase's word vectors are concatenated to form a sentence vector. A benchmark such as Euclidean distance is used to determine the degree of similarity between the two texts.

C. Leacock and M. Chodorow created an automated scoring system for short answer questions based on the predicate

argument structure; it provides 84 percent accuracy for pronominal reference, morphological analysis, and synonyms.
Using the Bayes theorem, L.M. Rudner and T.Liang developed an automated essay grader that achieves an accuracy of up to 80%. Automatic assessment of free text answers was developed by F. Noorbehbahani and a. a. Kardan by modifying the BLEU algorithm and calculating a similarity score between the student response and the reference response, which was more accurate than latent semantic analysis (LSA) and n-gram co-occurrence evaluation techniques.

III. METHODOLOGY

A. Overview

As inputs, we use neural networks to express phrases using word vectors learned independently from a large collection. We employ the Word2vec model (reconstructing linguistic contexts of words using two-layer neural networks) to generate the input word vectors, also known as Word Embedding) using skip gram method Formally, we consider a supervised learning situation in which each training sample is made up of a pair of fixed-size vector sequences (each x(ia),x(jb) Rdin) and a single label. The sequences may also be of varied lengths, with durations ranging from one example to the next. Our motivating example is the task of identifying similarity between phrases given sample pairings whose semantic similarity has been labelled as y by humans. Each x(ia) represents a word vector from the first sentence, whereas the x(jb) represents word vectors from the second sentence. As a result, we employ LSTMs to learn a semantic metric with the specific goal of learning a semantic metric.

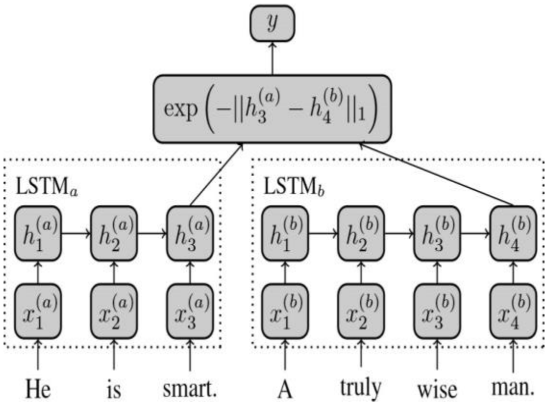


Fig. 1. Proposed Methodology overview I

B. Manhattan LSTM Model

Figure 1 depicts the proposed Manhattan LSTM (MaLSTM) model. We are primarily interested in Siamese architectures with connected weights because there are two networks, LSTMa and LSTMb, that each process one of the sentences in each pair such that LSTMa = LSTMb in this work. Nonetheless, the common untied version of this model may be more important for asymmetric domain applications such as information retrieval (where stored documents and search queries are stylistically distinct). The term MaLSTM (Manhattan LSTM) simply refers to the fact that they chose to compare the final hidden states of two regular LSTM layers using the Manhattan distance, this value is then compared to the teachers or evaluators graded Under the mean squared-error (MSE) loss function, value is the error signalling condition for LSTM and it back-propagates across time.

E. Training Details

Error% = Abs (Predicted value – Actual Value) x 100
A Actual Value_______

Fig. 3. Accuracy Function

There is only one hidden LSTM layer in the initial LSTM model, which is followed by a typical feed forward output layer.

C. Manhattan Stacked LSTM Model

The Stacked LSTM (MaSLSTM) is a better variant of this model since it has multiple hidden LSTM layers, each with a huge number of memory cells. Adding hidden layers to an LSTM model makes it more complex, earning it the title of deep learning technique. The performance of neural networks on a wide range of highly tough prediction problems is generally credited to their depth. To make a Multilayer Perceptron neural network more sophisticated, additional hidden layers can be added. The additional hidden layers' goal is to merge previously learnt representations and generate new ones at extremely high abstraction levels. From lines to shapes to objects, for example

Bidirectional Recurrent Neural Networks (RNNs) are a simple idea to grasp. It entails duplicating the initial recurrent layer of the network so that no two levels are contiguous; the input sequence is then fed directly to the first layer, while a reversed copy is fed to the second layer.

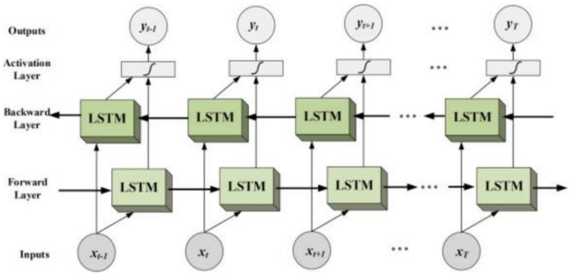


Fig. 2. Proposed Methodology overview II

D. Accuracy Function

we are calculating accuracy using basic function that is 100- total error %, where total error percentage is calculated by taking mean of all of error % of every value in the data set. Error % was calculated by Voice recognition This MALSTM (Manhattan LSTM) model is consists of 25-dimensional hidden representation ht and memory cells ct. Further Optimization of this model is done using Adam technique. We employ a validation set containing 20% of our SICK dataset. The results of the LSTM model are fed into the Manhattan Distance layer which predicts the score for the Answer and the Correct Answer. The loss used is Mean Squared Error (MSE).

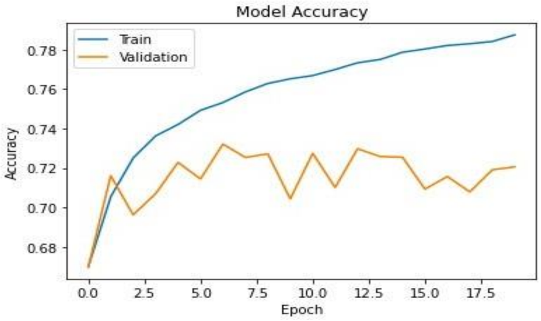
Our MASLSTM (Manhattan Stacked LSTM) model uses two LSTMs of 25- dimensional hidden representation h and memory cell which gave the best result on our test dataset among all the three proposed models. Its optimization is Adam and takes in 20% as Validation data of our SICK dataset. The results of the MASLSTM model are fed into the Manhattan Distance layer to predict the score between the Answer and the Correct Answer.

Our MABLSTM (Manhattan Bidirectional LSTM) model uses a single Bidirectional LSTM (due to limited labelled dataset) which gives the worst result on our test dataset among all the three proposed models. Its optimization is Adam, and it takes 20% of our SICK dataset as validation data. The results of the MABLSTM model are fed into the Manhattan Distance layer to predict the score between the Answer and the Correct Answer.

IV. RESULT ANALYSIS

We can see in all two graphs accuracy is gradually increas- ing with each epoch in both training and validation.

Fig. 4. Accuracy graph for model 1



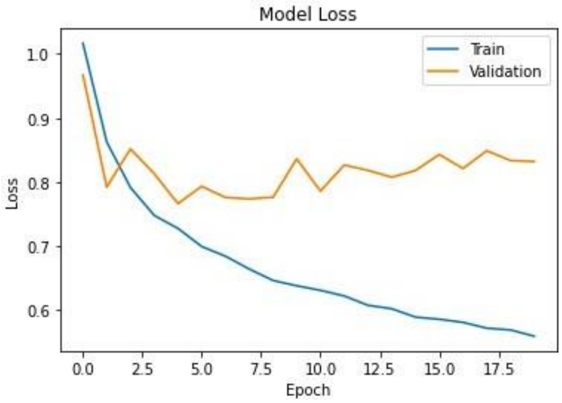


Fig. 5. Loss graph for model 1

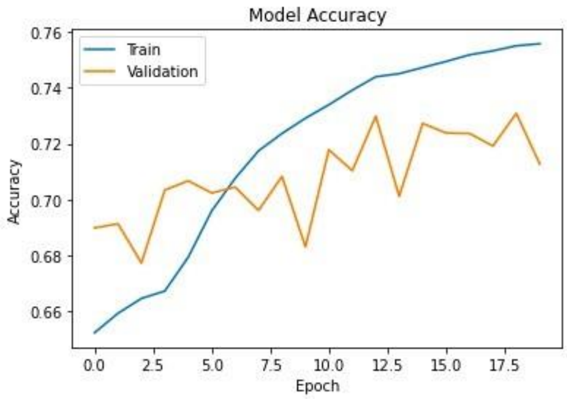


Fig. 6. Accuracy graph for model 2

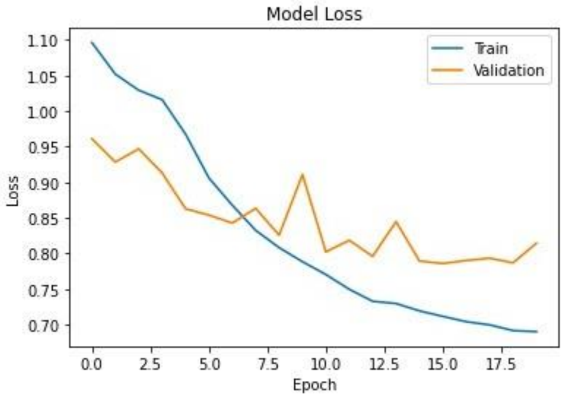


Fig. 7. Loss graph for model 2

V. CONCLUSION

The results of our research show that LSTM can simulate complex or challenging semantics if the representations are deliberately guided. We were able to circumvent the size limits of current labeled datasets by using synonym augmentation and pre-trained word embedding. According to the learned model, different hidden units are used to encode different traits and properties of each sentence. It was discovered that Stack LSTM performed the best of the three, providing much higher accuracy than the other two. Our approach can be used in real-time applications, such as analyzing subjective answer sheets, because it allows for efficient test-time inference. Because the methodology we've given uses pre-trained word-vectors as LSTM inputs, improvements in word embedding methods will undoubtedly benefit it, especially because these word-vectors capture synonymy and entity-relationships in greater depth. We also expect significant gains as the volume of labeled semantic similarity data grows, allowing for de novo word-vector learning tailored to our model, both for large sample sizes and statistical reasons.

VI. ACKNOWLEDGMENT

We are grateful to our project guide (Prof. Sharayu Lokhande) and HOD(Computer Engineering) Dr. Sunil Dhore
. This work has been done under Army Institute of Technology, Pune. We also thank all individuals who have helped and motivated the team in any way throughout the process. The authors take sole responsibility for the work.

REFERENCES

[1] Jonas Mueller, Aditya Thyagarajan. 2016, Siamese Recurrent Architec- tures for Learning Sentence Similarity, AAAI-16

[2] Neethu George, Sijimol PJ, Surekha Mariam Varghese "Grading De- scriptive Answer Scripts Using Deep Learning "Volume-8 Issue-5 March, 2019, ISSN: 2278-3075

[3] Piyush Patil 1, Sachin Patil 2, Vaibhav Miniyar3, Amol Bandal4 "Sub- jective Answer Evaluation Using Machine Learning" Volume 118 No. 24 2018 ISSN: 1314-3395

[4] Tanupriya Choudhury, Kartikeya Jain, Lakshya Aggarwal, Ayushi Gupta, Garv Saxena Computerized paper evaluation using neural network 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)

[5] Claudia Leacock and Martin Chodorow, C-rater: Automated Scoring of Short-Answer Questions.

[6] Ms. Sharmeen J. Shaikh, Ms. Prerna S. Patil, Ms. Jagruti A . Pardhe, Automated Descriptive answer evolution system using machine learning.

[7]L. M. Rudner and T. Liang, "Automated essay scoring using bayes' theorem," The Journal of Technology, Learning, and Assessment, vol.1, june 2002.

[8]F. Noorbehbahani and a. a. Kardan, The automatic assess- ment of free text answers using a modified BLEU algorithm, Comput. Educ., vol. 56, no. 2, pp. 337345, 2011.

[9]Salakhutdinov R., and Hinton, G., 2007 "Semantic hashing." in Proc. SIGIR Workshop Information Retrieval and Applications of Graphical Models.

[10]P.W. Foltz, W. Kintsch, and T.K. Landauer, "The Measurement of Textual Coherence with Latent Semantic Analysis," Discourse Processes, vol. 25, nos. 2-3, pp. 285-307, 1998

[11]J.L. McClelland and A.H. Kawamoto, "Mechanisms of Sentence Pro- cessing: Assigning Roles to Constituents of Sentences," Parallel Dis- tributed Process 2, D.E. Rumelhart, J.L. McClelland, and the PDP Research, eds., pp. 272-325, MIT Press, 1986.

[12]V. Hatzivassiloglou, J. Klavans, and E. Eskin, "Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning," Proc. Joint SIGDAT Conf. Empirical Methods in NLP and Very Large Corpora, 1999.

[13]V. Hatzivassiloglou, J. Klavans, and E. Eskin, "Detecting Similarity by Applying Leaning over Indicators," Proc. 37th Ann. Meeting of the Assoc. for Computational Linguistics, 1999.