IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 8, 2021, accepted January 13, 2021, date of publication January 25, 2021, date of current version February 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3054346

Automatic Short Answer Grading With SemSpace Sense Vectors and MaLSTM

CAGATAY NEFTALI TULU[1], OZGE OZKAYA[2], AND UMUT ORHAN[2]
[1]Information Technologies Division, Adana Alparslan Turkes Science and Technology University, 01250 Adana, Turkey
[2]Computer Engineering Department, Cukurova University, 01330 Adana, Turkey

Corresponding author: Cagatay Neftali Tulu (ctulu@atu.edu.tr)

ABSTRACT Automatic assessment of exams is widely preferred by educators than multiple-choice exams because of its efficiency in measuring student performance, lack of subjectivity when evaluating student response, and faster evaluation time than the time consuming manual evaluation. In this study, a new approach for the Automatic Short Answer Grading (ASAG) is proposed using MaLSTM and the sense vectors obtained by SemSpace, a synset based sense embedding method built leveraging WordNet. Synset representations of the Student's answers and reference answers are given as input into parallel LSTM architecture, they are transformed into sentence representations in the hidden layer and the vectorial similarity of these two representation vectors are computed with Manhattan Similarity in the output layer. The proposed approach has been tested using the Mohler ASAG dataset and successful results are obtained in terms of Pearson (r) correlation and RMSE. Also, the proposed approach has been tested as a case study using a specific dataset (CU-NLP) created from the exam of the ''Natural Language Processing'' course in the Computer Engineering Department of Cukurova University. And it has achieved a successful correlation. The results obtained in the experiments show that the proposed system can be used efficiently and effectively in context-dependent ASAG tasks.

INDEX TERMS Automatic short answer grading, MaLSTM, semspace sense vectors, synset based sense embedding, sentence similarity.

I. INTRODUCTION

Recently, pretrained language models such as BERT, GPT-2, ELMo [1]–[3] based on the processing of large corpora using advanced deep learning methods have taken much attention from Natural Language Processing (NLP) researchers. Thanks to these language models, it is possible to implement effective downstream NLP applications such as sentiment analysis, social virtual chat robots, or virtual smart robots that answer questions in a specific domain known as automatic question answering systems [4]–[7].

The automatic scoring of short answers in open-ended questions is one of the important studies of the NLP domain [8]. Multiple-choice exams are the most commonly preferred centralized examination models used in the world. In these exams, candidates are expected to answer questions by choosing one of the given options. Although it has been revealed that this approach is inadequate in measuring the skills and knowledge of the students [9], this method still dominates the centralized exam systems all over the world. One of the reason is; measurement and assessment process is very fast using optical readers and processing the results with software. Another reason is that if the exam type is open-ended which means answering the questions by writing arbitrary text and if the assessment is made through the human intervention, this may cause personal judgments to the answers and it affects the objectivity of the exam.

With rapid developments in natural language processing and machine learning applications, the idea of making exams based on open-ended questions that can be evaluated automatically when applied to a mass amount of students attracts educators. Generally, two assessment models are preferred for the automatic scoring of open-ended questions in the literature. While the evaluation in the first one is made based on some predefined criteria, in the second one, the assessment is based on semantic similarity between the correct answer and the student's response. In this method, the correct answer to the question is taken as a reference and compared with the student's answers in terms of semantic similarity between them.

For both assessment approach, deep learning-based models have the most successful results [10]. Especially the success of Long Short Term Memory (LSTM) [11] based models is remarkable. This method uses Manhattan vectorial similarity in the output layer. The words in the two sentences are

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

tokenized and transformed into the vectors and those vectors are given as input to twin LSTM networks. The semantic embeddings of the sentences are generated in the last hidden layers on the two networks and finally, by using Manhattan distance the semantic similarity of them is calculated to score the similarity. Mueller and Thyagarajan [12] have used Word2Vec [13] vectors in their study and have got the first rank in the semantic text similarity task (SICK) [14] of the SemEval 2014.

In the literature, one of the main success indicators of the Automatic Short Answer Grading (ASAG) studies is the high correlation and fewer loss values on the benchmark tests with publicly available and widely used datasets. The success of the proposed ASAG method with the Mohler dataset [40] is an important metric for the domain-dependent ASAG studies because Mohler dataset includes questions and answers in the field of computer engineering (domain-dependent), it is public and it consists of short answer sentences with maximum words (i.e. 20). Although some ASAG methods that adopt the sentence similarity approach measure the success of their studies using the SICK dataset [14], there are some shortcomings in the SICK dataset. Because the SICK dataset includes sentences whose semantic similarity is interpreted by leveraging different domains. It is not a specific domain dataset, not like a lecture notes that that is generated based on a chapter or a topic and not like a certain number of questions generated for the exam of a specific class. The domain-independent sentence similarity dataset as SICK is not preferred as benchmark data in ASAG studies because of the longer sentences than the school exams with short answers. According to the literature, classical unsupervised ASAG approaches used on the Mohler dataset [40] have low success values as shown in Table 4, and the expected success cannot be achieved in the new generation ASAG studies using advanced deep learning-based methods neither. The reason for this handicap might be the word representations and the deep learning architecture since they are the main components of deep learning-based studies. For this reason, it is proposed such a deep learning-based model should consist of two main components, one is the SemSpace sense vectors using concepts in WordNet [33], a well-defined knowledge-base created with the participation of scientists who are experts in linguistics and computational sciences, where only concepts are in a semantic relationship with each other. The second one is the Manhattan LSTM (MaLSTM) [12] network, which has been implemented on multi-layer LSTM networks, has proven its success in sentence similarity benchmark tests (i.e., SICK dataset). The training of the proposed system is built on the MaLSTM network. It is trained using the semantic similarity between the reference answer and the student's response.

High successes were gained in the tests performed on the CU-NLP[1] dataset that is specifically developed for this study. CU-NLP dataset consists of two exams and their answers for the Natural Language Processing course in fall 2019. The course is taken as a technical elective course by Cukurova University Computer Engineering undergraduate students. Then the benchmark tests of the proposed approach were performed on the Mohler dataset. Successful results on the Mohler dataset are an indicator that both the SemSpace sense vectors and the MaLSTM network provide better results together.

The rest of the paper is organized as related works in Section II. The data set used in the study, the SemSpace algorithm, determination of sense vector, and the MaLSTM method has been explained in Section III, the obtained results are revealed in Section IV by comparing results with different methods explaining the parameters, environment for the training of the proposed system, and finally the conclusion of the study and recommendations for the future studies are given in Section V.

II. RELATED WORKS

Page [16] made the first study of the computers to automatically evaluate student responses with natural language processing methods, it is known as Project Essay Grade. Page designed a system that automatically evaluates essays by using parameters such as text length, word number, word part of speech tag (POS Tag), and other parameters of the essays written by students. Burrows *et al.* [8] divided these studies on ASAG into several categories: the first category is known as concept mapping. The concept mapping idea is based on evaluating the student's answers by matching them with various concepts. ATM (Automatic Text Marker) [17] divides teacher and student responses into a list of minimal concepts, it counts the number of common concepts to calculate an assessment score. Perhaps the most important study in this category is the study known as c-rater [18], which aims to match as many concepts at the level of sentences as possible between reference answers and student responses. In this study, matching is done based on a set of rules as; syntactic variation, morphological variation, synonyms, and spelling correction rules. Wang *et al.* [50] compared three methods for grading earth science questions in secondary education, this comparison is made based on concept mapping, machine learning, and both. The initial concept mapping approach is based on the use of cosine similarity in the tf.idf (term frequency multiplied by the inverse document frequency) vectors by tackling bag of the word properties. The second concept mapping method was carried out by using a support vector machine (SVM) created by leveraging the bag of words properties. The last concept mapping approach is integrated with unigrams, bigrams, and speech bigrams, and a pure machine learning method using the SVM regression model.

Burrows names the second category of ASAG studies as Information Extraction techniques, information extraction techniques are aimed at extracting structured data from unstructured sources such as free text, and obtaining a feature set that represents structured data. WebLAS (Web-Based Language Assessment System) [19] identifies

[1] https://bmb.cu.edu.tr/uorhan/CuNLP.htm

important parts of the answers in disaggregated presentations and asks the teacher to approve and assign a weight to each. eMax [20] evaluation approach considers all possible formulations for the pattern matching. FreeText Author [21] provides a graphical user interface for teacher response entry and student response grading. Teacher answers are organized into syntactic-semantic templates for matching student responses. Auto-Assessor [22] focuses on standardized grading. Matches one-sentence student answers based on word-coordinate matching and synonyms with WordNet [33]. This refers to the matching of individual terms between teacher and student responses. In Auto-Assessor, each word that matches exactly is given a point, partial credit is given to words found from WordNet and associated with it, and the rest are not credited. In this way, an evaluation process is conducted.

According to Burrows, another category is corpus-based studies. Purpose of this category is the automatic evaluation of student responses with the statistical property data obtained from the corpus. In Atenea [24], while initially using BLEU [59] scale based on n-gram overlap and normalized sample length as the scoring method, Latent Semantic Analysis (LSA) [51] was also added to the evaluation pipeline. SAMText (Short Answer Measurement of Text) [23] is evaluated by applying an LSA variant based on an inverted index data structure that has been added to content from a web scan using related text on similar topics. Mohler and Mihalcea [52] have experimented with various approaches for the ASAG. They performed classification by comparing eight different semantic similarity measures, two of them are corpus based methods. These methods use Explicit Semantic Analysis (ESA) and LSA as the corpus based methods.

Machine learning-based methods are the another category of the ASAG studies. Machine learning-based methods typically use a set of features derived from text by using natural language processing techniques then combined into a single class or score using a classification or regression model. Features that include the bag of word and n-grams are typically used in this category. The e-Examiner [47] uses ROUGE metrics [54] as features required for machine learning. These are combined as linear regression. CAM (Content Assessment Module, Content Assessment Module) [48] uses the closest k-neighbor classifier and performs the assessment by measuring the overlap percentage of content at various linguistic levels between teacher and student responses. Madnani *et al.* [49], on the other hand, tackled eight different features, including BLEU, ROUGE, sentence number word link vectors, which are input to a logistic regression classifier and automatic evaluation is performed accordingly.

According to Burrows, the evaluation category is aimed to evaluate the methods previously developed using different methods and techniques for the ASAG. The aim is to apply certain metrics on the datasets given in the form of a competition to select the works with the highest success. The first of these is ASAP (Automated Student Assessment Prize), an automated ranking competition series organized by commercial competitive hosting company Kaggle. The Quadratic Weighted Kappa (QWK) method is used as a success criterion in ASAP competitions. Accordingly, the first three studies achieving the highest success are Tandalla [55], Zbontar [56], and Conort [57]. SemEval '13 Task 7 [58] is the first large-scale non-commercial ASAG competition. In this competition, the most successful works are known as Dzikovska '12 [25], Levy '13 [26], SoftCardinality [27], and UKP-BIU [28].

As a result of the rapid developments in deep learning methods, significant success has been gained in sentence-level semantic similarity studies developed using the LSTM network. In these methods, which are based on measuring the semantic relatedness of two texts, the ability of the neurons in the LSTM network to store the information formed in the previous step has also been a major factor. One of the most important works in this category is the MaLSTM [12] method, which measures the similarity of two sentences using Manhattan distance similarity at the output layer of the Siamese LSTM network architecture. Similarly, Othma *et al.* [29] conducted a study based on the principle of giving the closest semantic question and the relevant answer to the question asked by a user among the community questions and answers on the web for both English and Arabic, using Siamese LSTM and Manhattan vector distance. In benchmark tests performed on Yahoo Answers Dataset, successful results are obtained. Another study in this field is the study developed by Uto and Uchida [30], which includes a Deep Neural Network and IRT (Item Response Theory) along with an LSTM layer. Recently, several language models such as BERT [1], GPT-2 [2], ELMo [3] have been produced as a result of the processing of large corpora with deep learning techniques. These language models are pre-trained and made publicly available and they are used in the development of NLP applications with prior fine-tuning processes. ASAG studies have also been developed with these pre-trained language models. Zichao Wang et. al. [31], in their study named ml-BERT, showed that satisfactory results can be obtained as a result of using a limited number of labeled data sets specific to the domain of examination as training data. Chul Sung et. al. [32] have shown that by updating the pre-trained BERT language model with domain-specific books and question-answer data, better results can be achieved instead of fine-tuning the model.

III. MATERIAL AND METHOD

The approach proposed in this study consists of three fundamental steps. In the first step, the synsets and relationships in the WordNet 3.1 [33] data are arranged to train the SemSpace algorithm. After the training, a vector defined in Euclidean space is determined for each synset. In the second step, the datasets on which the tests will be carried out are separated into tokens and the correct synset candidates of these tokens are determined by the process of Word Sense Disambiguation (WSD). In the third step, the MaLSTM model is trained with the prepared data set. This section includes the preparation

process of the dataset used in the study, the details of the SemSpace algorithm and its sense vectors (1st and 2nd steps of the study), and the details of the MaLSTM model (3rd step of the study) are explained under separate headings in this section.

A. DETERMINING SENSE VECTORS WITH SEMSPACE METHOD

In the initial computational natural language processing applications, words were represented with one-hot encoding, but with rapid developments in big data which means a huge size of different types of unstructured data, the fact that such a costly vector representation could not be sustainable [36]. In order to bring the word vectors to an acceptable level in terms of both representation and dimension, the concept of word embeddings has been introduced. Word2Vec [13], GloVe [34], and FastText [35] can be listed as the widely used word embedding methods as of today. Although these methods are used successfully in many different applications, representing a word with more than one meaning (polysemous words) with a single vector creates serious meaning conflation deficiency problem [37]. Therefore, the term sense embedding has emerged. Although researchers have revealed several sense embedding methods [38], [39] a serious success could not be achieved in different downstream NLP applications. The SemSpace [15], a synset based contextualized sense embedding approach that aims to find a weight for each relationship and also a sense vector for each word defined in the WordNet. The SemSpace algorithm used in this study is slightly different from the SemSpace algorithm explained in detail in [15], in which a single vector representation for each synset is defined in WordNet. Besides, for WordNet relation weights, it was determined by a random search without aligning them to an expert intervened public word similarity dataset. For this purpose, a dataset representing all synsets and relationships in WordNet 3.1 data is prepared, and the SemSpace algorithm is run to determine each sense vector. As detailed in [15], SemSpace achieved successful results in the word level semantic similarity benchmark data sets (0.94 Spearman with RG65 [60] and 0.88 with MEN3000 [61]).WordNet 3.1 dataset is publicly accessible on the internet. There are 117K synset and 26 different types of 365K relations between synsets. For this study, all triples (node1, node2, relation) are collected into a text file. In the graph model of WordNet, nodes represent synsets. By running the SemSpace algorithm, the semantic relationships between neighboring nodes are calculated based on the Euclidean distance, and both weight and vector positions are obtained by a dual optimization approach. This algorithm is started with random initial vector positions and relation weights. Then by using the WordNet relations, similarities (weights) between vectors are adjusted with an iterative approach. The weights and the vectors that maximize the difference between Spearman correlation and MAE are stored as the optimum values. The similarity between the two vectors is calculated using (1).

$$\mathrm{Sim}(\mathrm{V}_1, \mathrm{V}_2) = e^{-\|\mathrm{V}_1 - \mathrm{V}_2\|}, \tag{1}$$

In (1), V1 and V2 represent the sense vectors in the Euclidean space. If the similarity value of the two vectors exceeds the relation weight between V1 and V2, the vectors are moved closer to each other, and vice versa, the vectors are moved away from each other. This is the change in the position of the vectors during the training of the SemSpace algorithm, this update is performed by (2)

$$\Delta = \eta(\mathrm{V}_1 - \mathrm{V}_2), \tag{2}$$

In (2), Δ represents the vector position updating value, while η shows the learning rate. The two relationships that connect three nodes in WordNet is transformed into two equations with three unknowns. Such infinite solution inconsistencies are referred to as the ''problem of lack of equations''. Although it is guaranteed that the three neighbor vectors in the SemSpace would be close to each other, it is stated that the vectors are poorly represented because the solution cannot be clarified.

Regarding the dimensions of the vectors, detailed analyzes have been made. It was observed that synsets that do not have semantic similarity with each other got closer to each other when the dimension of the vectors is selected relatively small (size $=$ 3). It has been observed that if the vector dimension is chosen large value (i.e. 300), the vectors initially are located far away from each other and only those with semantic relationships come closer to each other. Within the frame of the memory limits of the graphics card used, training was made using 300-dimensional vectors and obtained vectors have been used in this study. On the other hand, when the SemSpace algorithm is executed using the WordNet data, the weights of all relationship types are found around the value 0.5. Therefore, it is sufficient to determine the related synsets in newly defined relationships (regardless of the weights).

The Word Sense Disambiguation (WSD) process for the ambiguous words in the dataset is done as; first of all, the words belonging to student answers and reference answers are normalized by going through the steps tokenization, filter stop-words, and lemmatization as shown in Fig. 2. The context set is chosen by selecting the N tokens which are used most frequently in the dataset. Then each token is queried in the WordNet vocabulary list and converted to vectors using a lookup table. Those which were not found in WordNet are considered as the out of vocabulary (OOV) words. Those which have more than one synset candidate are determined by the WSD process. To do this, the candidate synset closest to the context cluster of the ambiguous word is selected. The mathematical representation of this WSD operation to find the best fit sense vector of dataset context is represented by (3).

$$C_{WSD} = \underset{G_j}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| C_j - P_i \right\|, \tag{3}$$

In (3), N is the number of synsets in the context cluster, $\mathrm{C_j}$ indicates the candidate synsets of the ambiguous word, and $\mathrm{P_i}$ is the sense vectors in the context cluster.

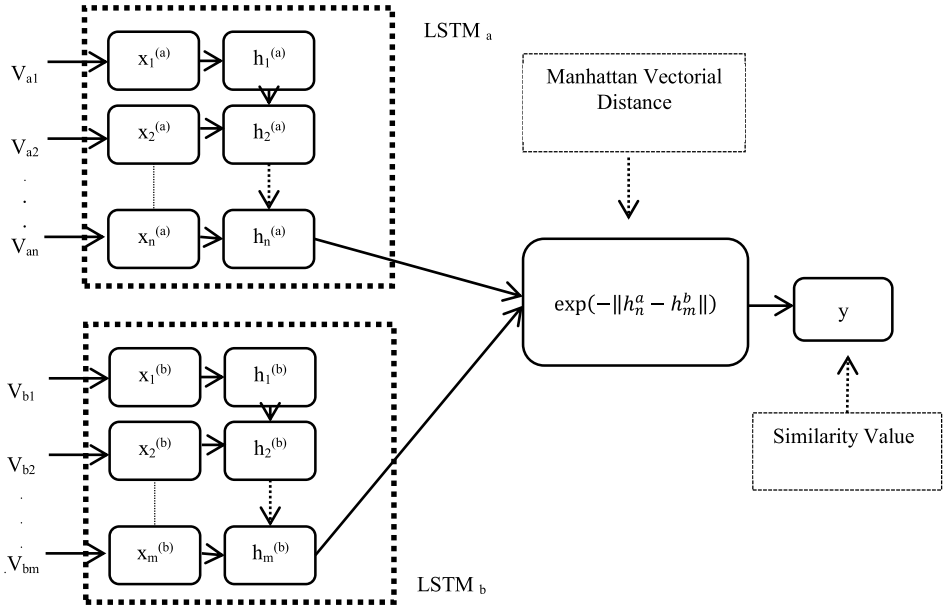

FIGURE 1. High-level overview of the proposed model.

B. GRADING WITH MALSTM

In the study, Manhattan LSTM (MaLSTM) [12] network, which is widely used in sentence similarity applications, has been preferred as a deep learning model. As seen in Figure 1, this network consists of two identical LSTM networks. While this network is being trained, the sense vectors of the words in the student's response text and the sense vectors of the words in reference answer text are fed into the input layer of the MaLSTM network. While the student response goes to the input layer of the $LSTM_a$, the reference answers are given to the input layer of the $LSTM_b$ as shown in Figure 1.

Each sentence pairs (student response and reference answer) entered to the LSTM networks with sense vectors is transformed into a sentence representation vector (a kind of contextualized text embedding) at the last hidden layer of each LSTM network, and the Manhattan distance of the text embeddings is calculated using vectorial similarity and it is normalized into a fixed interval [0,1]. While V_{ai} and V_{bi} in Figure 1 represent the sense vectors, h_n^a and h_m^b are the created contextualized text embeddings of the words in the second sentence. At the input layer, each word vector has a fixed length. Mueller and Thyagarajan [12] used Word2Vec [13] as the word embedding vectors in their studies. In this study, a synset based contextualized sense embedding method called as SemSpace and its sense vectors are used. Since SemSpace determines the representations of the concepts on WordNet, there must be a corresponding synset on WordNet for the words in the sentences to be compared. When a word cannot be matched any synset on WordNet, it is considered as OOV. In the last hidden layer of the LSTM network, the words belonging to the two sentences to be compared are aggregated and a single sentence vector representation of each sentence is created. The vectorial distance between the text vectors of these two sentences $(h_n)^a$ and $(h_m)^b$ is calculated using the Manhattan vector similarity.

The proposed system should have preliminary steps before going to the LSTM network. These steps are called preprocessing steps and the block diagram of these sequences is shown in Figure 2. Initially, the words belonging to the two sentences to be compared must be both tokenized, and stop word clearance should be taken place.

The obtained tokens should pass to the lemmatization process and the conceptual equivalents on WordNet should be found and if more than one semantic correspondence is found to the relevant word, a context-dependent WSD procedure should be applied as it is explained in section 3.A. After that, sense vectors are found using a lookup table from SemSpace. The deep learning stage of the proposed system starts in this step, the sense vectors to be the input to the LSTM network is converted into text vectors in the last hidden layer of the LSTM network. The similarity value found as a result of measuring the sentence representation vectors using Manhattan distance is compared with the real score given by the instructor for the training, and the MaLSTM network is trained according to the MSE (Mean Square Error) loss function and the weights in the hidden layers are updated by backpropagation.

C. THE ASAG DATASET USED IN THE STUDY

Two different datasets are used to test the proposed approach. The first one is the Mohler [40] dataset, it is widely preferred by most of the ASAG studies for benchmark purposes. Mohler's dataset is a domain-specific dataset generated by using Computer Science exams. The dataset consists of 12 exams and each exam consist of 7-8 questions. There are 87 reference answers in the dataset for 87 questions, each reference answer has student responses given by 26-31 students

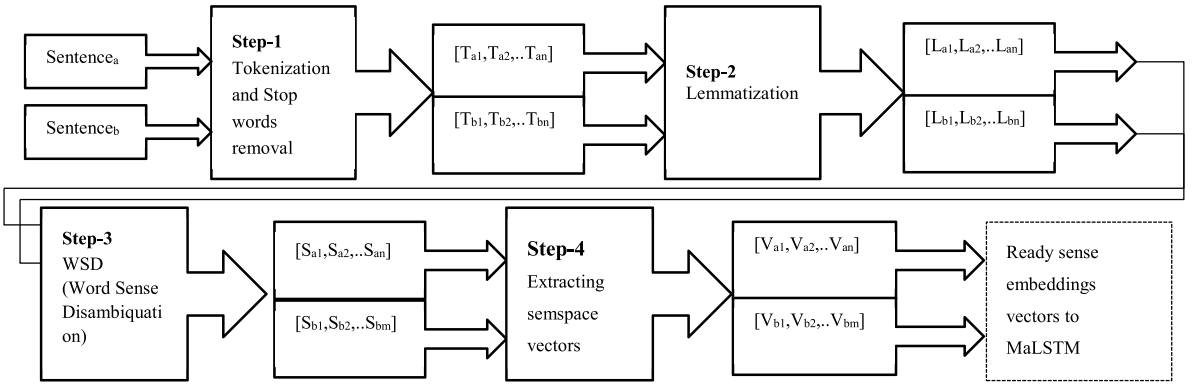

FIGURE 2. Preliminary textual preprocessing steps before entering into Neural Network.

TABLE 1. Structure of the training data.

Text1	Text2	Evaluation Score
Ref. Answer	Student answer1	Evaluation_score1
Ref. Answer	Student answer2	Evaluation_score2
…..	…….	…….
Student answer1	**Ref. Answer**	Evaluation_score1
Student answer2	**Ref. Answer**	Evaluation_score2

TABLE 2. Sample data from mohler training dataset.

Text1	Text2	Score
To simulate the behaviour of portions of the desired software product.	High risk problems are address in the prototype program to make sure…	0.7
To simulate the behaviour of portions of the desired software product.	To lay out the basics and give you a starting …	0.4
…….	……..	
High risk problems are address in the prototype program to make sure…	**To simulate the behaviour of portions of the desired software product.**	0.7
To lay out the basics and give you a starting ...	**To simulate the behaviour of portions of the desired software product.**	0.4

and evaluators score to each student responses by comparing the similarity of the reference answer and student response, evaluation scores are given from 1-5, where 5 is most relevant with reference answer and 1 is not relevant at all. Mohler Dataset is publicly available and can be downloaded from the internet. In our study, the Mohler dataset is transformed into csv file, each csv file consists of student responses and reference answer to one question. In this way, it is created 87 csv files, each csv file has a structure as; $< ref.\ answer >$, $< student\ answer >$, $< evaluation\ score >$. Since the used MaLSTM network consists of two identical LSTM layers. Each created each csv file is as in Table 1. By the way, each of the identical LSTM layer is trained with correct answers to have the same weights in its hidden layers.

Also, a new data set called as CU-NLP has been created specifically for this study, and used as the second dataset to evaluate the proposed model. The dataset prepared with the final exam conducted in the Natural Language Processing course, which is an elective course in the Department of Computer Engineering at Cukurova University for the fall

TABLE 3. Sample data from cu-nlp dataset.

Text1	Text2	Score
If it is a knowledge-based question, we can assume the question as a query…	This is the question answering problem. At first, I get tokens and	0.5
This problem can be solved by doing "word sense disambiguation",..	We can use a WordNet based disambiguation method. For this…	1.0
…….	……..	…
This is the question answering problem. At first, I get tokens and	**If it is a knowledge-based question, we can assume the question as a query…**	0.5
We can use a WordNet based disambiguation method. For this…	**This problem can be solved by doing "word sense disambiguation"…**	1.0

semester 2019. A total of 86 students took one exam each containing 2 open-ended questions via a web gui and 171 answer texts were recorded separately. One of the students has not answered one question. The reference answer text for each question was prepared by the instructor of the course. The evaluator gave a score between 0 and 100 for each student answer by performing the evaluation completely manually on the web gui, but it is normalized into the [0,1] interval in the study. The dataset prepared to train the system has two input texts (the first is the student's answer, the second is the expected correct answer) and a numeric output (the student's grade), the same approach has been applied as explained in Mohler's dataset, each reference answer and corresponding students responses have recorded into the same dataset file (csv) by crossing them each other. Sample data in the CU-NLP dataset as text1, text2, and grade are shown in Table 3.

Since the assessment of the student's responses was carried out completely manually, although the student's answer might be different from the answer expected by the instructor, the possibility of an alternative correct answer was considered and a second manual analysis was performed on the dataset by another instructor. Accordingly, some of the answers given by the students were presented as a keyword-based warning to the evaluator to prepare a new

alternative answer. So the evaluator verified the warning and prepared a second alternative answer text for a question and re-evaluated the student responses that came as a warning. Thus, it was confirmed that one answer text was found for one of the questions asked to the students, while two alternative answer texts could be correct for the second question. While cross-validating the study, the student answer texts in the dataset were compared with all alternative correct answers and the highest score obtained was considered as valid.

D. IMPLEMENTATION DETAILS

Initially, the SemSpace algorithm has been executed with WordNet 3.1 data and 300-dimensional sense vectors have been prepared for the total 117K synsets defined in WordNet. For the stop words clearance, tokenization, and lemmatization operations, the Python NLTK library is used. WordNet lemmatizer has been preferred since the sense vectors are generated using the WordNet vocabulary list. Also for the implementation of the proposed approach, the WordNet 3.1 data is downloaded and transformed into the text files to find the corresponding sense vectors. Sense vectors are the common inputs to feed the proposed system and they have been generated using the SemSpace algorithm. In order to solve ambiguity problems for the polysemy words, the WSD algorithm is implemented as explained in section 3.A. In order to build the proposed system, it is developed a python code by using TensorFlow to utilize the GPU power to save the training time. Keras is used to build a deep learning model. Input, Embedding, and LSTM layers are imported from Keras. Also, the sequential model is imported from Keras library to put embedding layers and LSTM layers inside this model. The custom layer which is known as the Manhattan Distance layer has been prepared as a separate function and added to the model, this layer takes two inputs from the previous layer. This custom layer tunes the text vectors received from two identical hidden LSTM layers.

Due to the nature of the model used, one more copy is prepared from all lines in the dataset file, and the places of the ''reference answer'' and ''student's response'' columns have been exchanged as shown in Table 1. Thus, two LSTM networks in the MaLSTM model have been twinned by learning the same sentences. The leave-one-out cross-validation method is used to determine the success on the dataset. In the training of the model, the mean square error was preferred as the loss function, Adam optimizer for optimization, and mean absolute error (MAE) as the success metric. The batch size is 1024 selected and 500 epochs for training. As a result of training the designed LSTM network, Pearson's r correlation values between the grades computed by the method and the grades given by the evaluator is calculated to compare the success of the method with other methods for the Mohler dataset. The LSTM layer embedding dimension is chosen as 60 and the input layer dimension as 300 for the SemSpace sense vectors. GPU Tesla T4 is used with compute capability: 7.5 provided by Google colab.

IV. RESULTS AND DISCUSSION

In order to test our method using the Mohler [40] dataset, 87 csv files are created for each exam question responses, and each of them is trained and tested independently. In most of the dataset files Pearson's (r) value$>$0.95 have been gained. As seen in Figure 3, Pearson values (Fig. 3.b) and MAE values (Fig 3.a) have a negative correlation, this shows us that the training process is completed consistently. The test results of each datasets are shown in terms of accuracy, there are 87 dataset files, each of them are trained separately, the x-axis in the Figure 3 shows the dataset file number and the y-axis shows the MAE (Mean Absolute Error) in Fig 3.a, Pearson correlation values (r) in Fig 3.b, and RMSE (Root Mean Square Error) in Fig 3.c. When Figure 3 is examined in-depth, the Pearson value for the dataset file 84 is 0.01. To find the root cause of this, dataset 84 file is manually checked. It is observed that all of the responses for each student have been scored with 1.0 except one that is given 0.9. Computed scores to the student responses for question 84 by the algorithm are exemined, it is observed that almost all of them are around 0.95, which is fine for the estimation but due to the nature of the data for question 84, this correlation value is acceptable. There are some lower Pearson values (around 0.6) for the dataset files like dataset file 43, 59, 63. These files are also manually examined and found that reference answers to the question are very short and also most of the student responses are scored 1.0 which is absolute point and response is counted as fully correct. Also, even student response is a longer sentence, it is scored as 1.0 by the evaluator just if the student response contains the reference answer words inside of the response text.

On the other hand, the proposed system is tested by inputting just one csv file that contains all the student responses to all questions. That file contains responses given to 87 questions by all students, that file has the records for the responses to all questions given by all students contains 4484 rows. Proposed system have trained and tested within this file. And found out %23 MAE and 0.15 Pearson correlation, this far from our expectation. Detailed analyzses are conducted to find the reasons of this deficiency. First of all, our proposed system generates a context set using words in the all vocabulary list generated from the dataset. When the number of the words in the context set increases, it causes slow determination of the correct synsets for ambiguous words and learning speed gets too slow and due to a large number of OOV words, learning hasn't completed as expected and this is one of the major reason for low success. As a result, to train the system just question by question as separate csv files is preffered, so responses to each question have been given to the system as input and a high success rate in terms of MAE and Pearson are obtained.

For the training and testing of the CU-NLP dataset, the same setup is used with the same model and with the same parameters 0.02 MAE and 0.989 Pearson correlation value have been obtained. Results have taken for this dataset also confirms that our model fits the expectations in terms of

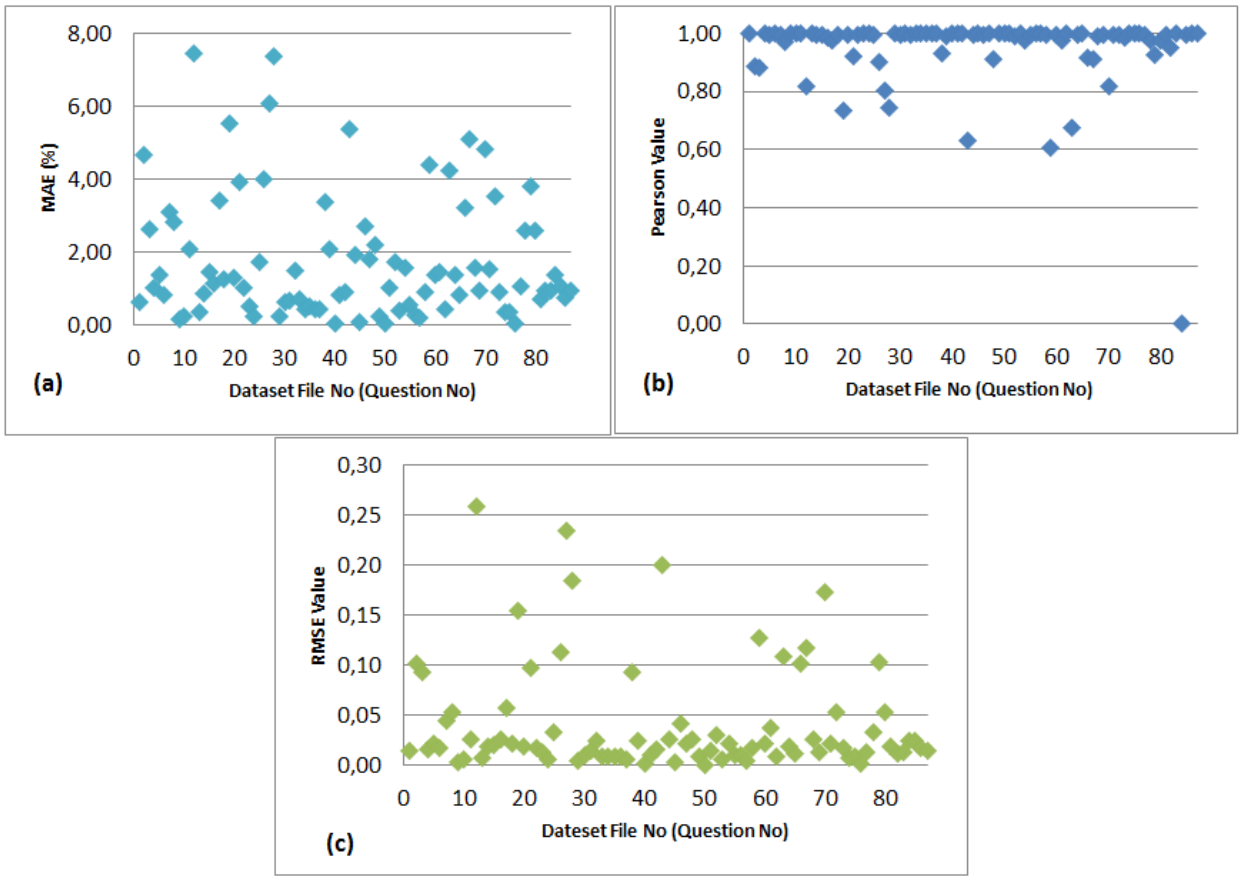

FIGURE 3. MAE (a), pearson (b), RMSE (c) results of each dataset files for the mohler dataset.

TABLE 4. Test results comparison of ASAG studies and our study in the literature using Mohler [40] dataset.

Study	Pearson's r	RMSE
Our Study[1]	0.949	0.040
Sultan et al. (2016)[2][41]	0.630	0.850
Saha et al. (2018)[2][42]	0.570	0.902
Ramachandran et al. (2015)	0.610	0.860
Mohler et al.[40]	0.518	0.978
Lesk [41]	0.450	1.050
tf-idf [41]	0.320	1.020

[1] Student responses to each question are trained separately, and the r score is the mean value of all training results.

accuracy, and also experiments made with this dataset shows the consistency of the model. Moreover, there are large number of OOV words in this dataset, but they are ignored and not evaluated and those OOV words did not much affect the performance and success of the proposed model. Most of them are typos and Turkish words entered by the students. In Figure 4, the training and test history of the model with the CU-NLP dataset in terms of model accuracy and loss according to the number of epochs can be seen. Graphs given in Figure 4 are generated using the CU-NLP dataset file that contains student responses to the CU-NLP exam questions.

By testing the proposed model with Mohler dataset, high accuracy in terms of Pearson correlation and RMSE as success criteria among the other studies have been determined. SemSpace sense vectors have positive contributions to the success of the approach proposed. Also training each question with corresponding student responses keeps the shorter context set and this allows faster training with higher accuracy. When the model is trained using all the student responses to all questions, the context set extends and training time gets too slow and low success and accuracy determined.

There is also SICK [14] dataset that sentence similarity-based ASAG studies are using for benchmark purposes. Our model is trained using the SICK dataset and got a 0.48 correlation value, the reason for the lower success within SICK dataset might be summarized as; SICK dataset consists of around 10K sentence pairs with semantic similarity values of the corresponding pairs. These sentences are randomly selected from daily life (newspapers, conversations from cinema films, etc.) and literary resources (books, poets…), and almost each sentence's context is independent of other one. But, Mohler and CU-NLP datasets have only one context for each dataset (Mohler and CU-NLP are about the Introduction to Computer Engineering and the Natural Language Processing courses, respectively). Also, in Mohler and CU-NLP dataset, the correct answer (reference answer) of each exam question, and the students' answers to that question are entered into the LSTM network one by one. Therefore, the semantic similarity value of each reference

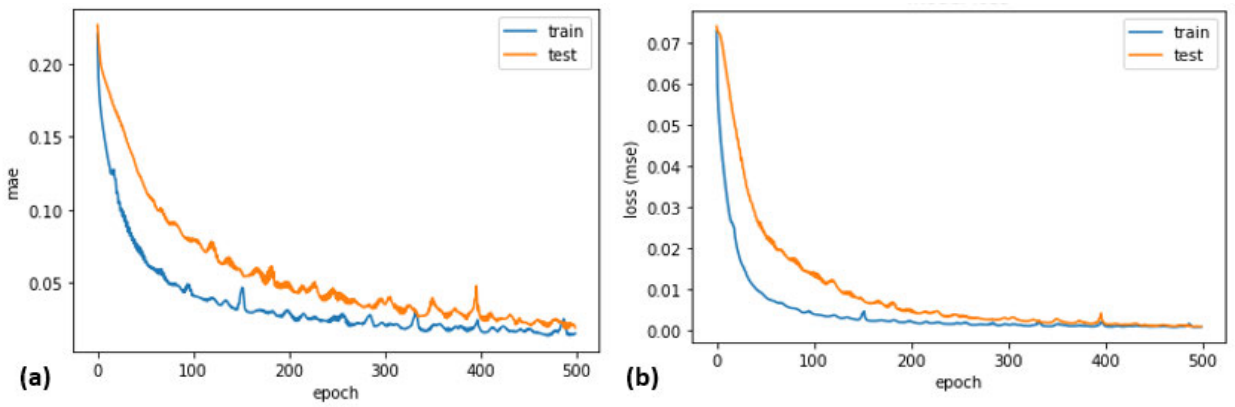

FIGURE 4. Plot of model MAE (a) & loss (b) on training/test for the CU-NLP dataset.

answer with 29 different sentences are given as input to the system, so the training is performed with many sentences and different similarity scores in the same context. By this way, the LSTM network is trained very well with the assessment of the sentences given to the reference answer. And the last reason is that the sentences of the SICK dataset is not eligible for the ASAG tasks because it contains too long (more than 30 words) sentences. But in the short answer grading concept, the responses within a max of 20 words are preffered. As a result, we can consider the SICK dataset as the gold standard for the sentence similarity tasks, but not for ASAG.

V. CONCLUSION

In this study, an automatic assessment approach for short answers is implemented using the MaLSTM network with two different datasets, one is the publicly available Mohler dataset which is widely used in ASAG studies as a benchmark, the another one is the CU-NLP dataset specifically generated for this task in Cukurova University Computer Engineering Department. The prominent aspect of the study is the use of sense representations obtained from concepts on the WordNet lexical-semantic network using the SemSpace method. The SemSpace algorithm generates sense vectors for each word sense defined on WordNet by using synsets and their relations. The study has gained a significant result by training it with the Mohler dataset. It also showed it's reliability and consistency within the training of special dataset CU-NLP used to test the proposed approach. It should be taken into account that not only the SemSpace sense vectors but also the MaLSTM model has made significant contributions to the success of the study.

As a future study, a model for automatic OOV handling using external corpora or external lexical semantic networks might be generated. By this way, OOV words can be added as the new synsets and they can be connected to available synsets with user-defined relations. Then, sense vectors for these new synsets can be computed by executing the SemSpace algorithm. Also, some misspelled words and typos are put into the OOV set and discarded in the training of the model, a sufficient method might be generated to allow automatic correction of the those words, and this decrease the size of the OOV vocabulary set and this makes positive contributions to the training success of the model. On the other hand, during the WSD process, the increase in both the number of words represented in the context set and the number of ambitious words causes highly increase in processing time. This is another shortcoming that is predicted to be solved by developing different perspectives in future studies.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805

[2] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskeve, "Language models are unsupervised multitask learners," OpenAI, San Francisco, CA, USA, Tech. Rep., 2019.

[3] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (Long Papers)*, vol. 1, 2018, pp. 2227–2237.

[4] S. Haitian, B. Dhingra, M. Zaheer, K. Rivard, R. Salakhutdinov, and W. Cohen, "Open domain question answering using early fusion of knowledge bases and text," 2018, *arXiv:1809.00782*. [Online]. Available: https://arxiv.org/abs/1809.00782

[5] Y. Mehmood and V. Balakrishnan, "An enhanced lexicon-based approach for sentiment analysis: A case study on illegal immigration," *Online Inf. Rev.*, vol. 44, no. 5, pp. 1097–1117, Jun. 2020, doi: 10.1108/OIR-10-2018-0295.

[6] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Commun. ACM*, vol. 59, no. 7, pp. 96–104, Jun. 2016, doi: 10.1145/2818717.

[7] S. Memeti and S. Pllana, "PAPA: A parallel programming assistant powered by IBM watson cognitive computing technology," *J. Comput. Sci.*, vol. 26, pp. 275–284, May 2018, doi: 10.1016/j.jocs.2018.01.001.

[8] S. Burrows, I. Gurevych, and B. Stein, "The eras and trends of automatic short answer grading," *Int. J. Artif. Intell. Edu.*, vol. 25, no. 1, pp. 60–117, Mar. 2015, doi: 10.1007/s40593-014-0026-8.

[9] Y. Oksuz and E. Demir, "Comparison of open ended questions and multiple choice tests in terms of psychometric features and student performance," *Hacettepe Univ. J. Edu.*, vol. 34, no. 1, pp. 259–282, 2019, doi: 10.16986/HUJE.2018040550.

[10] L. Galhardi, H. Senefonte, D. S. Thom, and J. R. Brancher, "Exploring distinct features for automatic short answer grading," in *Proc. Conf., Encontro Nacional de Inteligência Artif. Computacional*, 2018, pp. 1–12, doi: 10.5753/eniac.2018.4399.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[12] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2786–2792.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781

[14] M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, and R. Zamparelli, "SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 1–8, doi: 10.3115/v1/S14-2001.

[15] C. Tulu, "Semantic vector space model using Euclidean distance based relatedness: SemSpace," Ph.D. dissertation, Dept. Comput. Eng., Cukurova Univ., Adana, Turkey, 2019.

[16] E. B. Page, "The imminence of grading essays by computers," *Phi Delta Kappan*, vol. 47, no. 5, pp. 238–243, 1966.

[17] D. Callear and J. V. Jerrams-Smith ans Soh, "CAA of short non-MCQ answers," in *Proc. 5th Comput. Assist. Assessment Conf.*, M. Danson and C. Eabry, Eds. Loughborough, U.K.: Loughborough Univ., 2001, pp. 1–14.

[18] C. Leacock and M. Chodorow, "C-rater: Automated scoring of short-answer questions," *Comput. Humanities*, vol. 37, no. 4, pp. 389–405, 2003.

[19] L. F. Bachman, N. Carr, G. Kamei, M. Kim, M. J. Pan, C. Salvador, and Y. Sawaki, "A reliable approach to automatic assessment of short answer free responses," in *Proc. 19th Int. Conf. Comput. Linguistics*, 2002, pp. 1–4.

[20] D. Sima, B. Schmuck, S. Szoll, and A. Miklos, "Intelligent short text assessment in eMax," in *Towards Intelligent Engineering and Information Technology* (Studies in Computational Intelligence), vol. 243, I. J. Rudas, J. Fodor, J. Kacprzyk, Eds. Springer, 2009, pp. 435–445.

[21] S. Jordan and T. Mitchell, "E-assessment for learning? The potential of short-answer free-text questions with tailored feedback," *Brit. J. Educ. Technol.*, vol. 40, no. 2, pp. 371–385, Mar. 2009.

[22] L. Cutrone, M. Chang, and Kinshuk, "Auto-assessor: Computerized assessment system for marking Student's short-answers automatically," in *Proc. IEEE Int. Conf. Technol. Edu.*, Jul. 2011, pp. 81–88.

[23] O. Bukai, R. Pokorny, and J. Haynes, "An automated short-free-text scoring system: Development and assessment," in *Proc. 20th Interservice/Ind. Training, Simulation, Educ. Conf.*, 2006, pp. 1–11.

[24] E. Alfonseca and D. Perez, "Automatic assessment of open ended questions with a BLEU-inspired algorithm and shallow NLP," in *Advances in Natural Language Processing* (Lecture Notes in Computer Science), vol. 3230, J. Vicedo, P. Martínez-Barco, M. Muńoz, and S. Noeda, Eds. Berlin, Germany: Springer, 2004, pp. 25–35.

[25] M. O. Dzikovska, R. D. Nielsen, C. Brew, C. Leacock, D. Giampiccolo, L. Bentivogli, P. Clark, I. Dagan, and H. T. Dang, "SemEval-2013 task 7: The joint student response analysis and eighth recognizing textual entailment challenge," *Proc. 2nd Joint Conf. Lexical Comput. Semantic*, M. Diab, T. Baldwin, and M. Baroni, Eds. Atlanta, GA, USA: Association for Computational Linguistics, 2013, pp. 1–12.

[26] O. Levy, T. Zesch, I. Dagan, and I. Gurevych, "Recognizing partial textual entailment," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, H. Schuetze, P. Fung, and M. Poesio, Eds. Stroudsburg, PA, USA: Association for Computational Linguistics, 2013, pp. 451–455.

[27] S. Jimenez, C. Becerra, C. Universitaria, and A. Gelbukh, "SOFTCARDINALITY: Hierarchical text overlap for student response analysis," in *Proc. 2nd Joint Conf. Lexical Comput. Semantics*, vol. 2, M. Diab, T. Baldwin, M. Baroni, Eds. Atlanta, GA, USA: Association for Computational Linguistics, 2013, pp. 280–284.

[28] T. Zesch, O. Levy, I. Gurevych, and I. Dagan, "UKP-BIU: Similarity and entailment metrics for student response analysis," in *Proc. 17th Int. Workshop Semantic Eval.*, vol. 2, S. Manandhar and D. Yuret, Eds. Atlanta, Georgia: Association for Computational Linguistics, 2013, pp. 285–289.

[29] N. Othman, R. Faiz, and K. Smaïli, "Manhattan siamese LSTM for question retrieval in community question answering," in *Proc. Int. Conf. Ontologies, DataBases, Appl. Semantics*, 2019, pp. 661–677.

[30] U. Masaki and U. Yuto, "Automated short-answer grading using deep neural networks and item response theory," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2020, pp. 334–339.

[31] W. Zichao, S. L. Andrew, E. W. Andrew, P. Grimaldi, and R. G. Baraniuk, "A meta-learning augmented bidirectional transformer model for automatic short answer grading," in *Proc. 12th Int. Conf. Educ. Data Mining (EDM)*, 2019, pp. 1–4.

[32] C. Sung, T. Dhamecha, S. Saha, T. Ma, V. Reddy, and R. Arora, "Pre-training BERT on domain resources for short answer grading," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 6073–6077, doi: 10.18653/v1/D19-1628.

[33] C. Fellbaum, *WordNet: An Electronic Lexical Database* Cambridge, MA, USA: MIT Press, 1998.

[34] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[35] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2016, *arXiv:1607.04606*. [Online]. Available: http://arxiv.org/abs/1607.04606

[36] Q. Chen, Z. Xiaodan, L. Zhen-Hua, W. Si, and J. Hui, "Distraction-based neural networks for modeling document," in in *Proc. 35th Int. Joint Conf. Artif. Intell., IJCAI*, S. Kambhampati, Ed. New York, NY, USA: AAAI Press, Jul. 2016, pp. 2754–2760 [Online]. Available: http://www.ijcai.org/Abstract/16/391, 2016

[37] J. Camacho-Collados and M. T. Pilevar, "From word to sense embeddings: A survey on vector representations of meaning," *J. Artif. Intell. Res.*, vol. 63, pp. 743–788, Dec. 2018, doi: 10.1613/jair.1.11259.

[38] L. Qiu, K. Tu, and Y. Yu, "Context-dependent sense embedding," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 183–191.

[39] J. Li and D. Jurafsky, "Do multi-sense embeddings improve natural language understanding?" in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 683–693.

[40] M. Mohler, R. Bunescu, and R. Mihalcea, "Learning to grade short answer questions using semantic similarity measures and dependency graph alignments," *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2011, pp. 752–762.

[41] M. A. Sultan, C. Salazar, and T. Sumner, "Fast and easy short answer grading with high accuracy," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2016, pp. 1070–1075, doi: 10.18653/v1/N16-1123.

[42] A. Saha, V. Pahuja, M. Khapra, K. Sankaranarayanan, and S. Chandar, "Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph," in *Proc. 32nd AAAI Conf. Artif. Intell., (AAAI), 30th Innov. Appl. Artif. Intell. (IAAI-18), 8th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI)*, New Orleans, LA, USA, Feb. 2018, pp. 705–713.

[43] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.

[44] H. He, K. Gimpel, and J. Lin, "Multi-perspective sentence similarity modeling with convolutional neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1576–1586, doi: 10.18653/v1/D15-1181.

[45] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process. (Long Papers)*, vol. 1, 2015, pp. 1–11, doi: 10.3115/v1/P15-1150.

[46] W. Zhu, T. Yao, J. Ni, B. Wei, and Z. Lu, "Dependency-based siamese long short-term memory network for learning sentence representations," *PLOS ONE*, vol. 13, no. 3, 2018, Art. no. e0193919.

[47] C. Gutl, "e-Examiner: Towards a fully-automatic knowledge assessment tool applicable in adaptive e-learning systems," in *Proc. 2nd Int. Conf. Interact. Mobile Comput. Aided Learn.*, P. H. Ghassib Ed, 2007, pp. 1–10.

[48] S. Bailey and D. Meurers, "Diagnosing meaning errors in short answers to reading comprehension questions," in *Proc. 3rd Workshop Innov. Use NLP Building Educ. Appl. - EANL*, 2008, pp. 107–115.

[49] N. Madnani, J. Burstein, J. Sabatini, and T. O. Reilly, "Automated scoring of a summary writing task designed to measure reading comprehension," in *Proc. 8th Workshop Innov. Use NLP Building Educ. Appl.* J. Tetreault, J. Burstein, and C. Leacock, Eds. Atlanta, Georgia: Association for Computational Linguistics, 2013, pp. 163–168.

[50] H. C. Wang, C. Y. Chang, and T. Y. Li, "Assessing creative problem-solving with automated text grading," *Comput. Educ.*, vol. 51, no. 4, pp. 1450–1466, 2008.

[51] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, nos. 2–3, pp. 259–284, 1998.

[52] M. Mohler and R. Mihalcea, "Text-to-text semantic similarity for automatic short answer grading," in *Proc. 12th Conf. Eur. Chapter Assoc. Comput. Linguistics EACL*, 2009, pp. 567–575, doi: 10.3115/1609067.1609130.

[53] E. Gabrilovich and S. Markovitch, "Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge," in *Proc. 21st Nat. Conf. Artif. Intell.*, Boston, MA, USA: AAAI Press, vol. 2, 2006, pp. 1301–1306.

[54] C. Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. 1st Text Summarization Branches Out Workshop ACL*, Barcelona, Spain: Association for Computational Linguistics, 2004, pp. 74–81.
[55] L. Tandalla, "Scoring short answer essays," in *ASAP SAS Methodology Paper*, 2012. [Online]. Available: https://storage.googleapis.com/kaggle-competitions/kaggle/2959/media/TechnicalMethodsPaper.pdf
[56] J. Zbontar, "Short answer scoring by stacking," in *ASAP SAS Methodology Paper*, 2012. [Online]. Available: https://storage.googleapis.com/kaggle-competitions/kaggle/2959/media/jzbontar.pdf
[57] X. Conort, "Short answer scoring—Explanation of Gxav solution," in *ASAP SAS Methodology*, 2012. [Online]. Available: https://github.com/Gxav73/Gxav_Sol_ASAP_round2/blob/master/Gxav%20Description.docx?raw=true
[58] R. Nielsen, M. Dzikovska, C. Brew, C. Leacock, D. Giampiccolo, L. Bentivogli, P. Clark, I. Dagan, and H. Dang, "SemEval-2013 Task 7: The joint student response analysis and 8th recognizing textual entailment challenge," Assoc. Comput. Linguistics, Atlanta, GA, USA, Tech. Rep., 2013, pp. 263–274.
[59] D. Pérez and E. Alfonseca, "Application of the BLEU algorithm for recognizing textual entailments," in *Proc. 1st PASCAL Recognizing Textual Entailment Challenge*, 2005, pp. 9–12.
[60] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Commun. ACM*, vol. 8, no. 10, pp. 627–633, 1965.
[61] E. Bruni, N. K. Tran, and M. Baroni, "Multimodal distributional semantics," *J. Artif. Intell. Res.*, vol. 49, no. 1, pp. 1–47, Jan. 2014.

CAGATAY NEFTALI TULU received the B.S. and M.S. degrees from the Computer Engineering Department, Sakarya University, and the Ph.D. degree in computer engineering from Cukurova University. His research areas include natural language processing and machine learning.

OZGE OZKAYA received the B.S. degree in computer engineering from Cukurova University, where she is currently pursuing the M.S. degree. Her search areas include natural language processing and machine learning.

UMUT ORHAN received the B.S. degree from the Computer Engineering Department, Karadeniz Technical University, the M.S. degree in mathematics from Gaziosmanpasa University, and the Ph.D. degree in electrical and electronics engineering from Bulent Ecevit University. He is currently working with the Department of Computer Engineering, Cukurova University, as an Associate Professor. His research area includes natural language processing, machine learning, and embedded systems.