International Journal of Technology 12(3) 571-581 (2021)
Received January 2021 / Revised February 2021 / Accepted April 2021

International Journal of Technology

http://ijtech.eng.ui.ac.id

Automated Short-Answer Grading using Semantic Similarity based on Word Embedding

Fetty Fitriyanti Lubis[1*], Mutaqin[2], Atina Putri[2], Dana Waskita[3], Tri Sulistyaningtyas[3], Arry Akhmad Arman[1], Yusep Rosmansyah[1]

[1]*School of Electrical Engineering and Informatics, Smart City & Community Innovation Center, Institut Teknologi Bandung, Jl. Ganesa No.10, Kota Bandung 40132, Indonesia*
[2]*School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jl. Ganesa No.10, Kota Bandung 40132, Indonesia*
[3]*Faculty of Art and Design, Institut Teknologi Bandung, Jl. Ganesa No.10, Kota Bandung 40132, Indonesia*

Abstract. Automatic short-answer grading (ASAG) is a system that aims to help speed up the assessment process without an instructor's intervention. Previous research had successfully built an ASAG system whose performance had a correlation of 0.66 and mean absolute error (MAE) starting from 0.94 with a conventionally graded set. However, this study had a weakness in the need for more than one reference answer for each question. It used a string-based equation method and keyword matching process to measure the sentences' similarity in order to produce an assessment rubric. Thus, our study aimed to build a more concise short-answer automatic scoring system using a single reference answer. The mechanism used a semantic similarity measurement approach through word embedding techniques and syntactic analysis to assess the learner's accuracy. Based on the experiment results, the semantic similarity approach showed a correlation value of 0.70 and an MAE of 0.70 when compared with the grading reference.

Keywords: Automated grading; Short answer; Semantic similarity; Syntax analysis; Word embeddings

1. Introduction

In education, the assessment of learners is essential for evaluating their knowledge and understanding. Subjective assessment, such as short-answer questions, is the best choice to explore understanding and basic knowledge rather than objective assessment, such as multiple-choice or true/false questions. Short-answer questions require learners to respond by composing and integrating ideas expressed in their own sentences. However, grading short-answer exams has its challenges, especially in manual grading and with large-scale testing. It requires significant time and has problems in the consistency of the assessment. Automated scoring can be used as a feasible solution for the short-answer scoring process. As a solution, we adopted the sentiment analysis process, as in the studies of Santosh and Vardhan (2015), Mahadzir et al. (2018), and Surjandari et al. (2019).

Automatic short-answer grading (ASAG) is the process of evaluating this type of question response through a computer program by matching it with a related reference model (Sahu and Bhowmick, 2020).

* Corresponding author's email: fettyfitriyanti@staff.stei.itb.ac.id, Tel.: +62-22-2500985, Fax: +62-22-2500985
doi: 10.14716/ijtech.v12i3.4651

Unlike automated-essay scoring (AES), automated short-answer scoring (ASAS), another term used for ASAG, emphasizes the content rather than the style (Brew and Leacock, 2015). Therefore, a simple way of assessing short-essay answers is to measure the similarity of short-essay answers to an appropriate answer model. A combination of syntactic and lexical approaches will help the model determine the same semantic meaning in short-essay answers more simply.

Semantic similarity between learner answers (LA) and reference answers (RA) is the focus of many kinds of research related to ASAG (Mohler and Mihalcea, 2009; Mohler et al., 2011; Luchoomun et al., 2019). Three approaches to determine semantic similarity are knowledge-based, corpus-based, and word-embedding-based measures (Gomaa and Fahmy, 2013; Sahu and Bhowmick, 2020). Corpus-based similarity measures determine how many words are alike according to information obtained from large corpora (Gomaa and Fahmy, 2013). Latent semantic analysis (LSA) is the most popular corpus-based similarity technique. LSA assumes that words having close meanings will appear in similar segments of text. LSA uses the concept of a metaphorical "bag of words" that does not consider the actual order in gathering related words (Cutrone et al., 2011; Ratna et al., 2013).

Knowledge-based similarity measures determine how words are related using information derived from semantic networks (Gomaa and Fahmy, 2013). WordNet is the most popular semantic network in the field of measuring knowledge-based similarities among words. However, WordNet has inherent limitations related to the availability of qualified resources; they are not available for all languages, and proper names and domain-specific technical terms are underrepresented (Kenter and De Rijke, 2015).

The word-embedding model has shown successful results in representing words semantically in a vector space initially proposed by Mikolov and various colleagues (Mikolov et al., 2013a; Mikolov, et al., 2013b; see also, Bengio et al., 2003; Levy and Goldberg 2014). Word representation in a vector space reflects the semantics of the words. This paper proposes a semantic similarity calculation method based on this type of word-embedding for grading short-answer responses.

The following section reviews related work in automated short-answer scoring. Section 3 covers our proposed method. Section 4 reports on and analyses the experimental results. Finally, Section 5 presents the conclusion.

2. Related Work

Many studies have been conducted to assess the accuracy and reliability of ASAG. In general, there are two approaches used to automatically evaluate short answers. The first approach uses a supervised method (Roy et al., 2016; Sultan et al., 2016; Sahu and Bhowmick, 2020) that extracts features in the short answers. The second approach uses a variety of unsupervised methods to determine scores based on the distance between the learner responses and the answer model (Bin et al., 2008; Mohler and Mihalcea, 2009; Hasanah et al., 2018).

Sahu and Bhowmick (2020) conducted a comparative study of different features and regression models to improve ASAG. A set of text similarity features, such as knowledge-based measures, corpus-based features, and word-embedding features, were extracted for each pair of learner response and reference answer. Roy et al. (2016) proposed a transfer learning technique for ASAG that used an ensemble of a text and numerical classifier to reduce the continuous labeling effort needed for the task. Sultan et al. (2016) used a supervised model to predict scores based on semantic similarity and the correct answer between the learner response and reference answer.

Mohler and Mihalcea (2009) used unsupervised techniques for the task of ASAG by comparing some knowledge-based and corpus-based measures of text similarity. Gutierrez et al. (2013) proposed using a hybrid ontology-based information extraction (OBIE) system to identify correct and incorrect statements that combined extraction rules and machine learning-based information extractors. Sakaguchi et al. (2015) combined a learner answer- and reference answer-based approach, which included various features of both methods to build an automatic essay answer assessment model. Heilman and Madnani (2013) presented a system for short answer scoring that used stacking and domain adaptation to support the integration of various types of task-specific and general features.

3. Materials and Methods

3.1. Data Set

This study used the same dataset as the research of Hasanah et al. (2019). The daily test data for basic programming majors in a computer network engineering class (Vocational School Grade 2, SMKN 8 Semarang) were used. The aim was to compare the ASAG method in this study with the method proposed by Hasanah et al. (2019).

The dataset consisted of 224 learner responses (1 daily test × 7 questions × 32 learners). Each question in the dataset contained five reference answers with a different sentence structure, but we used only one reference answer. Table 1 shows the questions and the chosen reference answer. The test answers came from 32 learners who took the test and were collected manually.

Table 1 Questions and reference answers

No.	Question	Answer
1.	Apa yang kalian ketahui tentang algoritma? *(What do you know about algorithm?)*	Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis *(Algorithm is a sequence of logical steps for solving problems that are arranged systematically and logically)*
2.	Tuliskan algoritma untuk menyelesaikan masalah perhitungan luas persegi panjang *(Write the algorithm to find the area of a rectangle)*	Start, Masukkan panjang dan lebarnya, Hitung Luas panjang kali lebar, Hasil kali panjang dan lebar = luas, Finish *(Start, enter length and width, Calculate area length times width, Product length and width = area, Finish)*
3.	Apa yang kalian ketahui tentang flowchart? *(What do you know about flowchart?)*	Bagan (chart) yang menunjukkan alir (flow) di dalam program atau prosedur sistem secara logika *(A chart that shows the flow in a program or system procedure logically)*
4.	Ada berbagai macam simbol-simbol pada flowchart. Berfungsi untuk apa simbol decision pada flowchart? *(There are various kinds of symbols on a flowchart. What is the function of the decision symbol on a flowchart?)*	Pemilihan proses berdasarkan kondisi yang ada. *(Process selection based on existing conditions.)*
5.	Berfungsi untuk apa simbol terminator pada flowchart? *(What is the function of the terminator symbol on a flowchart)*	Simbol terminator digunakan untuk permulaan (start) atau akhir (finish) dari suatu kegiatan. *(The terminator symbol is used for the start or end of an activity.)*
6.	Apa yang dimaksud dengan inisialisasi variabel? *(What is variable initialization?)*	Inisialisasi variabel adalah mengisi nilai untuk pertama kalinya ke dalam variabel.

		(Variable initialization is filling a value for the first time into a variable.)
7.	Apa fungsi operator pada pemrograman? *(What are the functions of operators in programming?)*	Operator berfungsi untuk memanipulasi nilai dari suatu variabel *(Operators function to manipulate the value of a variable)*

Two instructors graded each answer independently using an integer scale from zero (completely false) to four (perfect solution). Table 2 shows the scores assigned by the two instructors to the responses of four learners on Question 7. As can be seen from Tables 1 and 2, the dataset contains the various questions, reference answers, learner responses, and both instructors' scores for each learner response. To determine the correlation value and mean absolute error (MAE), the instructors' scores and their average (representing manual grading) were compared with the automatic answer grading.

Table 2 Example learner responses and instructors' manual grading for question 7

No.	Learner Answer	Score 1*	Score 2*	The Avg. Score
7.1	operator adalah simbol simbol khusus yang digunakan untuk mengoperasikan suatu nilai data (operand) *(Operator is a special symbol used to operate a data value (operand))*	4	4	4
7.2	berfungsi untuk mengoperassikan suatu data *(Function to operate a data)*	4	4	4
7.3	operator digunakan untuk memanipulasi atau melakukan proses perhitungan pada suatu nilai variable. *(Operators are used to manipulating or performing calculations on a variable value.)*	4	4	4
7.4	fungsi operator = mengelola segala bentuk pemrograman *(operator function = manage all forms of programming)*	2	1	1.5

*Score 1 from instructor one and Score 2 from instructor two

3.2. Proposed Method

Figure 1 shows the flow of the proposed ASAG research solution. The ASAG process consists of three modules: preprocessing, which prepares the text; word embedding, which is a word vector generator; and scoring, which handles the actual assessment. Word embeddings like word2vec (i.e., word to vector representation) do not pay attention to the arrangement of words in a sentence (Xu and Ye, 2017). For example, if a learner's answer is *"Indonesia adalah ibu kota Jakarta*" (Indonesia is the capital of Jakarta) and the reference answer is *"Jakarta adalah ibu kota Indonesia*" (Jakarta is the capital of Indonesia), the learner's answer is erroneous because it mistakes the two names; however, it will have an almost identical semantic or even similarity score when the sequence of the words are ignored.

The initial stage of ASAG is the preprocessing of the learner responses and reference answers. The next step is the process of calculating the value of the short answers using the assessment module. The assessment module utilizes word vector representations created by the word vector generator module.

The short answer is expressed in sentences or paragraphs that could differ from reference answers lexically or syntactically but have the same semantic meaning. The learner responses are scored based on the semantic similarity with the reference answers.

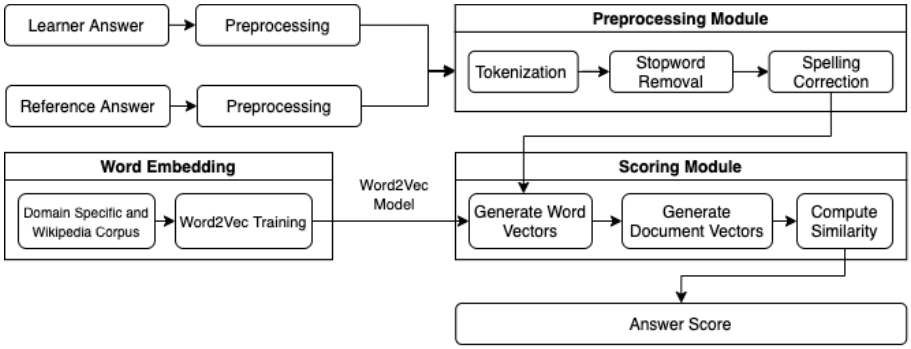

Figure 1 Overview of our ASAG solution

Before calculating the LA and RA's semantic similarity, we first represent each LA document and RA as vectors. The average sentence vector represents the document vector. We use the method proposed by Arora et al. (2016) and apply it at the document level.

Algorithm 1 determines the document vectors based on sentence and word vectors. The word vectors are obtained by word-embedding models previously trained from a large corpus using a continuous bag-of-words (CBOW) method or skip-gram. Each sentence in each LA document and RA is converted into a sentence vector by calculating the average word vector that composes the sentence. The average calculation uses a weight, $\alpha /(\alpha + p(w))$, where α is constant, and $p(w)$ is the probability of the word frequency in a document. Finally, the document vector is obtained by calculating the average of the sentence vectors that compose the form.

Algorithm 1:		Document Embedding
Input:		Word embeddings $\{v_w : w \in V\}$, a set of sentences S, parameter a and estimated probabilities $\{p(w) : w \in V\}$ of the words
Output	:	Document embeddings v_D
1	:	**for all** sentence s in S **do**
2	:	$v_s \leftarrow \frac{1}{\lvert s \rvert} \sum_{w \in S} \frac{\alpha}{\alpha + p(w)} v_w$
3	:	**end for**
4	:	Form a matrix X whose columns are $\{vs : s \in S\}$, and let u be its first singular vector
5	:	**for all** sentence s in S **do**
6	:	$v_s \leftarrow v_s - uu^T v_s$
7	:	**end for**
8	:	$v_D \leftarrow \frac{1}{n} \sum v_s$

3.3. Experimental Design

This section presents our experimental design with text preprocessing, word embedding generation, short-answer scoring, and testing scenario. Figure 1 shows the method in detail.

3.3.1. Text preprocessing

Text preprocessing plays a role in preparing text data that is suitable for processing in the module for calculating the score of the short-answer responses. This module processes the learner responses and reference answers. The output is a list of selected words (i.e., "bag of words"). This module performs the following processes: text normalization, tokenization, stop word removal, stemming (removing the suffix from a word and reducing

it to its root word), part-of-speech (POS) tagging, and dependency parsing using the Stanford NLP code.

3.3.2. Word embedding generation

The semantic similarity between an LA and RA is determined by calculating the cosine similarity of their respective document vectors. To obtain the document vector or document embedding, we need to first obtain the word embedding or word vector. In this experiment, we used the word2vec training method to obtain the word vectors. Because training word2vec requires a large corpus, we used the Wikipedia corpus as a universal word-embedding model. We selected the full Wikipedia dump in Bahasa (Indonesia) that contained about 365,939 article lines.

To train word2vec, we used the Gensim Python library. We set the CBOW model training and context window size to five with trained word vectors having 400 dimensions. Finally, we obtained about 98,000 word vectors from the Wikipedia corpus.

3.3.3. Scoring module

The scoring module calculates the final value of the short answers based on the word embedding-based semantic similarities between the learner's responses and the reference answers. The module also calculates sentences containing negations between the learner's response and connections with the answer. In addition, the process used syntactic analysis, POS tagging, and dependency parsing.

For the syntactic analysis, we used universal dependency languages for Indonesian treebanks (https://universaldependencies.org/#download), which were converted from the universal dependency treebank v2.0 (legacy) (Green et al., 2012). The syntactic analysis process took three steps. In the first step, we determined the POS for each word in the sentence; for example, the word *'mahasiswa'* (students) is a noun, and the word *'belajar'* (study) is a verb. In the second step, we determined the relationship of each word in the sentence by building a parsing tree with the tags using dependency parsing. For example, in the sentence *'mahasiswa belajar'* (students study), the word *'mahasiswa'* (students) has a relationship with the word *'belajar' (study)* as a nominal subject (nsubj). This method provides flexibility even when the order of words is changed [like *'mahasiswa belajar'* (students study) or *'belajar mahasiswa'* (study students)]. In the last step, we linked each word in the sentence with the word vector from the word-embedding model. The word vector was then used to calculate the semantic similarity.

3.3.4. Semantic similarity calculation

We measured the semantic similarity between the LAs and RAs to determine the learner's accuracy based on the similarity value. This value determines the distance between the document vector value, the learner's response, and the reference answer. For example, a document d is composed of several words w with the word vector v_w, so that the calculation of the document vector value V_d is through the following equation:

$$V_d = \frac{\sum_{i=1}^{n} v_{w_i}}{n} \quad (1)$$

Measuring the semantic similarity between the learners' answers (A) and reference answers (R) can be obtained by measuring the document vector distance (V_d) with V_{d_A} equal to the learner answer document vector, and V_{d_R} equal to the reference answer document vector. The calculation of the distances between the vector documents can use the cosine similarity equation as follows:

$$Sim\ (A,R) = \frac{V_{d_A}.V_{d_R}}{\|V_{d_A}\|.\|V_{d_R}\|} = \frac{\sum_{i=1}^{n} v_{d_{Ai}}.v_{d_{Ri}}}{\sqrt{\sum_{i=1}^{n} v_{d_{Ai}}^2}\ .\sqrt{\sum_{i=1}^{n} v_{d_{Ri}}^2}} \quad (2)$$

3.3.5. Checking the meaning of a sentence containing a negation

A learner's answer with semantic similarities to the answer reference can still negate the sentence found in that reference answer. The meaning of two types of negation sentences can be determined by *syntactic analysis* through analyzing the sentence by utilizing the sentence's grammatical structure.

The grammatical structure of a sentence can be determined by using the POS tagger and dependency parser. The POS tagger is used to identify the word class in a sentence. In contrast, the dependency parser is used to analyze the words based on their dependencies. This study decomposed the sentences into the POS and dependencies using the universal dependency languages for Indonesian treebanks.

3.4. Short Answer Scoring and Testing Scenario

We determined the short essay's score by calculating the semantic similarity value between the learner's answer and the reference answer, $sim(A,R)$. We then multiplied it by the proportion of sentence pairs with opposite meanings, t, to obtain the following equation for the short essay's score, $score(A)$:

$$score(A) = sim(A,R).t \quad (3)$$

The testing scenario consisted of three stages:

1) Conversion of the data set to CSV format with columns arranged question, reference answers, and learner's responses.
2) Deletion of words or phrases in the learner's response and connection to the solution was part of the question.
3) Normalization of symbols or abbreviations.

The second stage aim was that the learner's answers and the answer references did not contain the words or terms in the questions, as shown in Table 3. Furthermore, the third stage aimed to translate symbols or signs into a word or phrase. After the third stage, each learner's answer obtained a final score. The final score was a discrete number with a value range from zero to four, as shown in Table 4. The output was then saved in CSV format as data for the next testing process.

Table 3 Examples of questions and learner answers that contain the same phrases as the questions

Question (Q)	Apa fungsi operator pada pemrograman? *(What functions do operators perform in programming?)*
Reference Answer (RA)	Operator berfungsi untuk memanipulasi nilai dari suatu variabel *(Operators function to manipulate the value of a variable)*
Learner Answer (LA) (1)	operator adalah simbol simbol khusus yang digunakan untuk mengoperasikan suatu nilai data (operand) *(Operator is a special symbol used to operate a data value (operand))*
Learner Answer (LA) (3)	operator digunakan untuk memanipulasi atau melakukan proses perhitungan pada suatu nilai variable. *(Operators are used to manipulating or performing calculations on a variable value.*

Table 4 Examples of automatic essay answer assessment results for learners' responses (see Table 2)

Question and Learner No.	Grading from instructor 1	Grading from instructor 2	Average instructors grading	Grading from automatically
7.1	4	4	4	3
7.3	4	4	4	3

4. Results and Discussion

The evaluation of the ASAG system used semantic similarity based on word embeddings. The instructors' manual scoring was compared with the automated scoring generated by our approach (see No. 2–4 in Table 5). We used the Pearson correlation coefficient (r) and the MAE as the evaluation measures.

The Pearson correlation coefficient indicates the degree of strength of the linear relationship between, for example, the instructor ratings of the short answers and the automatic grading of the same answers. Correlation values close to one indicate a strong relationship between the manual and automated evaluation; a correlation value of zero indicates no relationship. Negative values in this case are problematic, as this means that the higher-rated answers of one set are the lower-rated in the other and vice versa. The equation for calculating the correlation coefficient is as follows:

$$r_{xy} = \frac{n \sum_{i=1}^{n} x_i y_i - (\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i)}{\sqrt{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \sqrt{n \sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}} \quad (4)$$

where x is a score produced by one method, y is a score for the same answer produced by a second method, and n is the total number of learner answers.

The MAE is a metric that can be used to compare two assessment methods. In addition, it can also stand on its own as an error measure of an individual method. The MAE is calculated as follows:

$$MAE = \frac{\sum_{i=1}^{n} |x_i - y_i|}{n} \quad (5)$$

We used the correlation coefficient and MAE to make a number of grading comparisons regarding our test dataset (Table 5). All the comparisons are based on the seven questions to which the learners responded. Based on the evaluation results for Comparison No. 4, the proposed research method obtained a correlation coefficient value (r) of 0.7085 with the averaged instructors' scores. This value indicates a strong relationship between the proposed automated scoring and the manual scoring that was conducted. Also, the level of accuracy of the automated essay answer assessment, MAE, was 0.7009.

The *MAE* calculated between the two instructors' scores (shown as Comparison No. 1 in Table 5) was relatively low (0.2768) because both instructors had quite similar grading scores. However, the *MAE* comparisons for all the automated gradings were more than 0.7 because they used word embedding and syntactic analysis. For example, the word *"membuka/pembuka"* (open/opener) in a learner's answer corresponding to the beginning word in the reference answer had a low similarity value of 0.2258. In the word2vec training corpus, the word *"membuka/pembuka"* and the word *"permulaan"* (start/beginning) were not used in the same context; therefore, this increased the *MAE* values.

For automatic grading with the approach of Hasanah et al. (2019) (shown as Comparison No. 5 in Table 5), we report the correlation and MAE scores from their paper. From Table 5 (Comparison No. 4), our approach compared to the same instructor average scores had a correlation score of 0.7085 and 0.7009 for the MAE score. Thus, our approach has been shown to anticipate various answers from learners using just one reference answer. Our approach does not perfectly replicate the instructors' scores but does so better than the previous approach, as demonstrated by the respective correlation coefficients.

Table 5 Results of the ASAG evaluation

No.	Grading Score Comparison Made	Correlation (*r*)	Mean Absolute Error (*MAE*)
1	Instructor 1 (Manual) vs. Instructor 2 (Manual)	0.8964	0.2768
2	Instructor 1 (Manual) vs. **Proposed ASAG** (Automated Using Word Embedding and Syntactic Analysis)	0.6788	0.7232
3	Instructor 2 (Manual) vs. **Proposed ASAG** (Automated Using Word Embedding and Syntactic Analysis)	0.6932	0.7679
4	Instructor Average (Manual) vs. **Proposed ASAG** (Automated Using Word Embedding and Syntactic Analysis)	**0.7085**	**0.7009**
5	Instructor Average (Manual) vs. Previous ASAG [Automated, as Reported by Hasanah et al. (2019)]	0.6542	0.9499

5. Conclusions

This paper explored the semantic similarity approach for automatic short answer grading. We believe this paper made two significant contributions. First, while the previous research used multiple answer references, our proposed method used only a single reference answer. Second, to make our method more influential than the previous study, we applied syntactic analysis by utilizing POS tagging, dependency relationships, and the word-embedding method. In the future, we intend to improve the word2vec model by adding more text corpora as training model input. Furthermore, we would like to expand the research problem, especially for short essay answers requiring a sequence of solutions.

Acknowledgements

This research was jointly funded by the Indonesian Ministry of Research, Technology and Higher Education under the WCU Program managed by the Bandung Institute of Technology and the Research, Community Service, and Innovation Program (P2MI) of the Faculty of Arts and Design, Bandung Institute of Technology (ITB). This study is also partially funded by MIT-Indonesia Research Alliance (MIRA) IMPACT, to whom the authors are grateful. It was also supported by the Smart City & Community Innovation Center, Bandung Institute of Technology (ITB).

References

Arora, S., Liang, Y., Ma, T., 2016. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. *In:* International Conference on Learning Representations, pp. 416–424

Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C., 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, Volume 3, pp. 1137–1155

Bin, L., Jun, L., Jian-Min, Y., Qiao-Ming, Z., 2008. Automated Essay Scoring using the KNN Algorithm. *In:* Proceedings—International Conference on Computer Science and Software Engineering, CSSE 2008, pp. 735–738

Brew, C., Leacock, C., 2013. Automated Short Answer Scoring Principles and Prospects. *In:* Handbook of Automated Essay Evaluation, Current Applications and New Directions, Routledge, pp. 136–152

Cutrone, L., Chang, M., Kinshuk, 2011. Auto-Assessor: Computerized Assessment System for Marking Student's Short-Answers Automatically. *In:* Proceedings IEEE International Conference on Technology for Education, T4E 2011, pp. 81–88

Dandibhotla, T.S., Vardhan, B.V., 2015. Obtaining Feature and Sentiment Based Linked Instance RDF Data from Unstructured Reviews using Ontology Based Machine Learning. *International Journal of Technology,* Volume 6(2), pp. 198–206

Gomaa, W., Fahmy, A., 2013. A Survey of Text Similarity Approaches. *International Journal of Computer Applications,* Volume 68(13), pp. 13–18

Green, N., Larasati, S.D., Žabokrtský, Z., 2012. Indonesian Dependency Treebank: Annotation and Parsing. *In:* Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation, pp. 137–145

Gutierrez, F., Dou, D., Martini, A., Fickas, S., Zong, H., 2013. Hybrid Ontology-Based Information Extraction for Automated Text Grading. *In:* International Conference on Machine Learning and Applications, pp. 359–364

Hasanah, U., Astuti, T., Wahyudi, R., Rifai, Z., Pambudi, R.A., 2018. An Experimental Study of Text Preprocessing Techniques for Automatic Short Answer Grading in Indonesian. *In:* Proceedings—Third International Conference on Information Technology, Information Systems and Electrical Engineering, pp. 230–234

Hasanah, U., Permanasari, A.E., Kusumawardani, S.S., Pribadi, F.S., 2019. A Scoring Rubric for Automatic Short Answer Grading System. *Telkomnika (Telecommunication Computing Electronics and Control)*, Volume 17(2), pp. 763–770

Heilman, M., Madnani, N., 2013. ETS: Domain Adaptation and Stacking for Short Answer Scoring. *In:* SEM 2013—Second Joint Conference on Lexical and Computational Semantics, 2(SemEval), pp. 275–279

Kenter, T., De Rijke, M., 2015. Short Text Similarity with Word Embeddings. *In:* Proceedings International Conference on Information and Knowledge Management, pp. 1411–1420

Levy, O., Goldberg, Y., 2014. Linguistic Regularities in Sparse and Explicit Word Representations. *In:* Proceedings of the Eighteenth Conference on Computational Natural Language Learning, pp. 171–180

Luchoomun, T., Chumroo, M., Ramnarain-Seetohul, V., 2019. A Knowledge Based System for Automated Assessment of Short Structured Questions. *In:* IEEE Global Engineering Education Conference, pp. 1349–1352

Mahadzir, N.H., Omar, M.F., Nawi, M.N.M., 2018. A Sentiment Analysis Visualization System for the Property Industry. *International Journal of Technology*, Volume 9(8), pp. 1609–1617

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., 2013a. Distributed Representations of Words and Phrases and Their Compositionality. *In*: Proceedings of the 26th International Conference on Neural Information Processing Systems–Volume 2, pp. 3111–3119

Mikolov, T., Yih, W., Zweig, G., 2013b. Linguistic Regularities in Continuous Space Word Representations. *In*: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751

Mohler, M., Bunescu, R., Mihalcea, R., 2011. Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments. *In:* ACL-HLT 2011—Proceedings of the 49th Annual Meeting of the Association for Computational

Linguistics: Human Language Technologies, pp. 752–762
Mohler, M., Mihalcea, R., 2009. Text-to-Text Semantic Similarity for Automatic Short Answer Grading. *In:* Proceedings EACL 2009—12th Conference of the European Chapter of the Association for Computational Linguistics, pp. 567–575
Ratna, A.A.P., Artajaya, H., Adhi, B.A., 2013. GLSA Based Online Essay Grading System. *In:* Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering, pp. 358–361
Roy, S., Bhatt, H.S., Narahari, Y., 2016. An Iterative Transfer Learning Based Ensemble Technique for Automatic Short Answer Grading. *ArXiv*, *abs/1609.0*
Sahu, A., Bhowmick, P.K., 2020. Feature Engineering and Ensemble-Based Approach for Improving Automatic Short-Answer Grading Performance. *IEEE Transactions on Learning Technologies*, Volume 13, pp. 77–90
Sakaguchi, K., Heilman, M., Madnani, N., 2015. Effective Feature Integration for Automated Short Answer Scoring. *In:* NAACL HLT 2015—2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, pp. 1049–1054
Santosh, D.T., Vardhan, B.V., 2015. Obtaining Feature and Sentiment Based Linked Instance RDF Data From Unstructured Reviews using Ontology Based Machine Learning. *International Journal of Technology,* Volume 2, pp. 198-206
Sultan, M.A., Salazar, C., Sumner, T., 2016. Fast and Easy Short Answer Grading with High Accuracy. *In:* 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016–Proceedings of the Conference, pp. 1070–1075
Surjandari, I., Wayasti, R.A., Zulkarnain, Laoh, E., 2019. Mining Public Opinion on Ride Hailing Service Providers using Aspect Based Sentiment Analysis. *International Journal of Technology*, Volume 10(4), pp. 818–828
Xu, X., Ye, F., 2017. Sentences Similarity Analysis based on Word Embedding and Syntax Analysis. *In:* 2017 17th IEEE International Conference on Communication Technology Sentences, pp. 1896–1900