Automated Short Answer Scoring Using Weighted Cosine Coefficient

Feddy Setio Pribadi[1,2], Teguh Bharata Adji[1], Adhistya Erna Permanasari[1]
[1]Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Indonesia
[2]Department of Electrical Engineering, Universitas Negeri Semarang, Indonesia
feddy.setio.p@mail.ugm.ac.id, adji@ugm.ac.id, adhistya@ugm.ac.id

***Abstract*— Scoring short answer question (consisting of at most 100 words) requires a special treatment, particularly on the term weighting process. With such limitation, it is therefore infeasible to have term weighting with term frequency model because the frequency of term appearance is very rare. This paper proposes a new method, named Weighted Cosine Coefficient. The experiment result shows that Weighted Cosine Coefficient method can improve the correlation value on Automated Short Answer Scoring (ASAS) system compared to that of Jaccard Coefficient, Dice Coefficient, Cosine Coefficient, Weighted Jaccard Coefficient, and Weighted Dice Coefficient. The average improvement of the correlation value on the tested data set is 0.56.**

Keywords—term based, short answer scoring, subjective test, Weighted Cosine Coefficient, Weighted Jaccard Coefficient, Weighted Dice Coefficient, Jaccard Coefficient, Dice Coefficient, Cosine Coefficient.

I. INTRODUCTION

Evaluating students in order to assess their achievement is considered important in a learning process. Such assessment will measure students' ability in receiving information given by the teachers during learning process. The use of information technology in such evaluation process creates a possibility for measuring learning outcomes in a large scale testing, within a short time and a consistent way. There are two types of test to assess students' achievement, namely objective test and subjective test. Some experts argue that complex thinking cannot be measured by utilizing objective questions [1][2]. Complex thinking requires students to solve a certain case according to their level of knowledge. Such test can only be performed by using subjective test. It means a type of test, which requires students to answer using their own sentences.

Some automatic scoring system for subjective test have been developed. Those scoring systems are categorized into two types, which are AES and ASAS. AES (Automated Essay Scoring) is a system for automatic correction for answer in a long essay form, while ASAS (Automated Short Answer Scoring) is a system for automatic checking for short essay answer [3]. This paper focuses its discussion on the 2nd system: ASAS system.

Automatic scoring process for subjective test questions is basically designing a computer program to be able to measure similarity rate between sentences, namely between Reverence Answer that has been prepared by question maker and Student Answer that student provides. There are several previous researches attempting to solve such problem with various methods. Some tried to employ string based method [4][5][6], knowledge based method [7], and some even investigated 3 methods to find the best among them, which are string based, knowledge based, and corpus based methods [8][9].

Corpus based ASAS system usually requires vast amount of supporting data to compute similarity value between words [10]. In addition to that, corpus based and knowledge based ASAS system generally only deal with a certain domain and requires frequent updates. In other words, if both ASAS system are to be employed for another domain, then a new set of corpus is required to be set in advance [6].

The problem of short answer question data set is that it has limited number of words. Short answers generally consist of one phrase up to three or four sentences [11], with maximum of 100 words for each sentence [12]. In order to solve such issue, this paper proposes term based method that disregards term frequency, and thus it is very relevant to deal with sentence that has only a limited number of words.

II. RELATED WORKS

Several ASAS systems that have been developed in previous researches—particularly those that use string based methods—will be discussed in this paper.

IndusMarker is designed for assessing short answer question [11]. In IndusMarker, Reference Answer is in the pattern of keywords, which is organized in an XML structure, referred as QAML (Question Answer Markup Language). IndusMarker limits the types of question that it can handle, and therefore automatic scoring of this system would be more optimal. Type of question that IndusMarker can handle includes short description, comparison of items, definition of term, exemplification and listing of items.

In IndusMarker, the structure of reference answer needs to be set up manually based on its set of keywords. The accuracy in arranging the keywords on QAML structure requires a high level of carefulness since this arrangement that will determine the score assigned to the students.

978-1-4673-9060-6/16/$31.00 ©2016 IEEE

Another research that uses term based method in ASAS system is Modified Bleu (M-Bleu). It is a formula defined by Noorbehbahani [5] that is applied to calculate similarities between sentences by considering keywords in Reference Answer. Every word in Reference Answer has different weight, based on its role in giving important meaning to the sentence. This is the reason for weighting each word with different weight. M-Bleu formula is as following:

$$M - BLEU_{ra} = \exp\left[\sum_{i=1}^{N} w_n \log(WP_{ra}(n))\right] \quad (1)$$

where WP_{ra} is the n-gram weight and W_n is 1/n (n-gram precession). Equation (1) is used to measure similarity between student answer and provided reference answer. This research aims to assign term weighting for word or key phrase (n-gram) in provided reference answer. This term weighting is performed in order to increase the similarity value between student answer and reference answer. After computing similarity value between student answer and reference answer, the final score for the student is determined with the following formula:

$$\text{sim}(s,r) = \lambda \times \text{BP}_r \times \text{M} - \text{BLEU} + (1-\lambda) \times \text{S}_0 \quad (2)$$

where BP_r is Brevity Penalty on reference answer, r is the reference answer, s is the student answer, and λ is the weight for common words order. The value of λ is assigned to be 0.85, obtained from experiment result.

Ben Omran et al [4] conducted experiment that implements combination of character-based measurement and term based measurement. Such research attempted to measure similarity between the meaning of reference answer and that of student answer, by considering 3 items: Longest Common Subsequence (LCS), Common Words (COW), and Semantic Distance (SD). Such similarity value between student answer and reference answer is computed with the following formula:

$$\text{sim}(S_1,S_2) = \lambda_1 \times \text{sim}_{\text{lcs}}(S_1,S_2) + \lambda_2 \times \text{sim}_{\text{cow}}(S_1,S_2) + \lambda_3 \times \text{sim}_{\text{sd}}(S_1,S_2) \quad (3)$$

$Sim(S_1, S_2)$ is the similarity score between reference answer (S_1) and student answer (S_2) and is also the total of LCS, COW and SD score. λ is the weight given for each similarity value on those three measures. The values of weight are determined by experiment of 100 times of combination on weight for LCS, COW and SD. The optimal weights for this research—after some experiment—are λ_1=0.1, λ_2=0.4 and λ_3=0.5.

Sankar [7] conducted a research that uses similar data set with the data that is used in this paper. Sankar implemented knowledge based method for his research by developing an ontology. Ontology is used for mapping between terms contained in student answer and those in reference answer. The matching process begins by parsing the reference answer and student answer. Stanford Parser is implemented to mark location of words within the compared sentence. After getting keywords, those terms are then arranged in a graph data structure. The similarity value between two sentences can later be computed from the graph.

Kudi [13] developed an online evaluation system by utilizing text mining algorithm for short text. Text mining algorithm is used for getting important information within a sentence, which is in a form of keywords. The keywords are then later arranged in the form that is proposed in [11]. Instead of using XML—as Siddiqi did, Kudi chose to use JSON that he used to simplify the XML structure and to make the system development easier since it is based on JavaScript.

Auto-Assesor is a system developed by Cutrone [14]. In this system, the main focus is developing canonical form (the desired formal form). Auto-Assesor requires several preprocess steps to have the data in canonical form, including punctuation removal, stop words removal and stemming. Auto-Assesor uses principle from Natural Language Processing (NLP) to measure similarity between student answer and reference answer.

This paper proposes a novel approach that differs with previous researches mentioned earlier. This paper uses dataset with only a single reference answer and with limited number of words in both reference answer and student answer. The proposed method refers to how an evaluator performs the scoring mechanism to the student answer. An evaluator generally only pays close attention to important words on student answer and then compares them to the context in the reference answer. Based on such scoring mechanism, every word that forms a certain sentence can be assigned with different weight. It is because not every word in the student answer is significant term that the evaluator expects to appear in the answer. Term weighting process in this paper does not use probability as performed in [5], but it uses binary term vector. The matching of student answer and reference answer is done via computation of coefficient value, not by rule based method that was done earlier by [7][11][13]. The determination of student's final score is also done in a simpler fashion compared to that of [4][5]. The proposed method is an advanced modification of Cosine Coefficient formula, referred as Weighted Cosine Coefficient (WCC).

III. PROPOSED METHOD

This research uses data set published by Mihalcea, from web.eecs.umich.edu/~mihalcea/download.html page, which was also used by previous researchers to do their works [7][8][9]. This research uses only 300 data set, selected from available 2610 dataset. The data are obtained from the 10 first questions only, in which each question is answered by 30 students. The proposed ASAS system is developed as web based application by employing PHP programming language. In order to analyze the performance of the proposed method, the computed similarity value between student answer and reference answer of this method will be benchmarked with that of five other methods, namely Jaccard Coefficient, Dice Coefficient, Cosine Coefficient, Weighted Jaccard Coefficient, and Weighted Dice Coefficient. Table 1 shows the original

formula of Dice, Jaccard, and Cosine Coefficient methods [15].

TABLE I. FORMULA OF DICE, JACCARD AND COSINE COEFFICIENT

Similarity Coefficient (R,S)	Actual Formula
Dice Coefficient (DC)	$\frac{2\lvert R \cap S\rvert}{\lvert R\rvert + \lvert S\rvert}$
Jaccard Coefficient (JC)	$\frac{\lvert R \cap S\rvert}{\lvert R\rvert + \lvert S\rvert - \lvert R \cap S\rvert}$
Cosine Coefficient (CC)	$\frac{\lvert R \cap S\rvert}{\lvert R\rvert^{1/2} \cdot \lvert S\rvert^{1/2}}$

The matching process of student answer and reference answer for Jaccard Coefficient, Dice Coefficient and Cosine Coefficient will be explained in the following, in which Table II shows part of the dataset that will be used for such explanation.

TABLE II. SAMPLE DATASET

Question	What is the role of prototype program in problem solving?
Reference Answer (RA)	to simulate the behaviour of portion of the desire software product
Student Answer (SA)	A program that simulate the behaviour of portion of the desire software product

Table II shows a reference answer and one of student answer whose organization of words between the two is very identical. From those two sentences, the binary vector for RA∩SA = "simulate, behavior, portion, of, the, desire, software, product;" not SA = "to;" and not RA = "a, program, that" From the binary vector computation, after previous tokenization process, the value of RA∩SA is 8 and the length of sentence of RA=11 and SA is 13 each.

By using formula as described in Table I, Jaccard Coefficient gives similarity value of 8/((11+13)-8)=0.5; the computed score by using Dice Coefficient formula is (2*8)/(11+13)=0.67; and the computed score from Cosine Coefficient is 8/(sqrt(11)*sqrt(13))=0.67. Those 3 coefficient formula can have a maximum value of 1 that will happen if the two compared sentences have identical arrangement of words.

Weighted formula is developed to improve the similarity value score between the two compared sentences. From the computing illustration in the previous paragraph, the computed score cannot reach beyond 0.9, which means that similarity value between those two sentences has not reached 90%. The formula of Weighted Cosine Coefficient (4), Weighted Jaccard Coefficient (5), and Weighted Dice Coefficient (6) are follow:

$$\frac{\sum_{i=1}^{n_r}\sum_{j=1}^{n_s}\left(W(R_i \cap S_j)\right)}{\sqrt{\sum_{i=1}^{n_r} W.R_i} \cdot \sqrt{\sum_{j=1}^{n_s} W.S_j}} \quad (4)$$

$$\frac{\sum_{i=1}^{n_r}\sum_{j=1}^{n_s}\left(W(R_i \cap S_j)\right)}{(\sum_{i=1}^{n_r} W.R_i + \sum_{j=1}^{n_s} W.S_j) - (\sum_{i=1}^{n_r}\sum_{j=1}^{n_s}(R_i \cap S_j)} \quad (5)$$

$$\frac{2\left(\sum_{i=1}^{n_r}\sum_{j=1}^{n_s}\left(W(R_i \cap S_j)\right)\right)}{\sum_{i=1}^{n_r} W.R_i + \sum_{j=1}^{n_s} W.S_j} \quad (6)$$

Where W is weighted for keywords. The words that become a keyword have an important position or meaning of the sentence. R is a reference answer, S is student answer.

For several methods, such as LSA (Latent Semantic Analysis) and Cosine Similarity, the weighting process is conducted by computing term frequency and document frequency. This is possible for long sentence, in which a term can appear several times within a sentence. However, this is not the case for short sentence, a term may only appear once in a sentence. Therefore, the weighting process in WCC is done manually by the evaluator. Terms/words that are considered as keywords of the sentence will have higher weight compared to others.

The weight is determined based on its importance on the reference answer. This weight assigning method is also conducted by Noorbehbahani. Noorbehbahani categorized terms into 4 types, which are very important = 4, important = 3, fairly important = 2, and not important 1. Terms in this paper is categorized into 5 types, as shown in Table III below.

TABLE III. CATEGORY AND TERM WEIGHTING

Category	Weight
Higly important	5
Very important	4
Important	3
Fairly important	2
Not important	1

The decision to use 5 categories is based on experimental research. After testing with 3 categories, 4 categories, 5 categories, and 6 categories, the term weighting with 5 categories produces the most optimal term weighting.

Table IV shows weighting process on the reference answer. In this process, each term is assigned a certain weight depending on its importance. Terms that are considered important are given higher value because they are the ones that are expected to appear in student answer. Meanwhile, those that are not significant are assigned with smaller values. Some researches even omit conjunctions to increase the performance of the system [16][9]. An example of a computation to measure similarity value of sentences from Table II by using Weighted Cosine Coefficient (equation 4) is shown below.

TABLE IV. TERM WEIGHTING ON REFERENCE ANSWER

Term	Weight
simulate, behavior, software	5
Portion, desire	4
product	3
of, the	1

The values of parameter on WCC formula in Table II after token are following: R∩S=8, R=9 and S=11. The weight of each term is given on Table IV, and therefore the WCC value is computed as follows: ∑W. (R∩S) = (5*3) + (4*2) + (3*1) + (1*2) = 28, ∑W.R = (5*3) + (4*2) + (3*1) + (1*2) + 1 = 29, ∑W. S = (5*3) + (4*2) + (3*1) + (1*2) + 1 + 1 + 1 = 31. WCC = 28 / (√29 * (√31) = 0.93

Measurement using M-Bleu formula (equation 1) which is each n-gram weighting = 4, and therefore the M-Bleu value is computed as follows: $WP_{ra(1\text{-}gram)}$ = (4*8) / (4*11) = 0.7, $WP_{ra(2\text{-}gram)}$ = (4*8) / (4*10) = 0.8, $WP_{ra(3\text{-}gram)}$ = (4*7) / (4*9) = 0.8, $WP_{ra(4\text{-}gram)}$ = (4*7) / (4*8) = 0.9. The computed score from M-Bleu is = 0.90. The result still lower than value that obtain of Weighted formula Coefficient.

From the computation, it is shown that by using Weighted Cosine Coefficient, the score increases to be 0.93. In other words, the similarity value between the two compared sentences is 93%.

In this paper compares only similarity coefficient values produced by a method based on the calculation of 1-gram (unigram). The proposed method is expected to have a good performance with faster computing

IV. RESULT AND DISCUSSION

The performance of the proposed method is measured by comparing it with the result of Jaccard Coefficient (JC), Weighted Jaccard Coefficient (WJC), Dice Coefficient (DC), Weighted Dice Coefficient (WDC), and Cosine Coefficient (CC). The result of the comparison is shown on Table V. The table shows correlation value of the score produced by the system and the score given by the evacuator. The evaluator score is obtained from the average of the score given by two evaluators.

TABLE V. CORRELATION OF SEVERAL METHODS

No question	JC	WJC	DC	WDC	CC	WCC
1	0.74	0.82	0.76	0.86	0.80	0.87
2	0.48	0.65	0.48	0.64	0.60	0.66
3	0.19	0.15	0.18	0.13	0.23	0.16
4	0.26	0.56	0.29	0.67	0.33	0.69
5	0.30	0.35	0.28	0.32	0.34	0.35
6	0.64	0.74	0.64	0.73	0.60	0.71
7	0.50	0.60	0.59	0.72	0.62	0.74
8	0.51	0.65	0.55	0.69	0.57	0.70
9	0.40	0.45	0.40	0.45	0.40	0.45
10	0.24	0.28	0.25	0.29	0.24	0.28

According to Table V, the highest correlation is obtained from the first question with the value of 0.87 and the lowest value is on the third question with the value of 0.15. The highest correlation is obtained when the score of the system is similar to the score produced by evaluator.

From the experiment, it is observable that the proposed method can produce almost similar score with the one produced by evaluator on certain type of questions. Some of them are questions about definition of a term, question that asks to mention something that is definitely a process. and question that asks to mention the location of certain syntax within a source code. Those questions tend to ask the students to answer objective questions, because the answer is definitive. With such condition, the difference between student answer and reference answer is not significant. Therefore, by focusing on giving weight on keywords, the proposed method can give optimal score. However, this is not the case for type of question that asks student to mention the difference between two items and type of question that asks student the reason why certain event occurs. All of the methods (proposed method and the other 3 previous methods) cannot produce similar scores given by evaluator. This is because the sentence that student writes varies significantly. Another failing point of those methods in giving good score is when reference answer is shorter than student answer. It is because the value of numerator will be significantly larger compared to the value of denominator.

The proposed method is very accurate for ASAS system that deals with subjective definite questions. One type of subjective definite questions is recalling question that asks students to answer a question on certain context, in which the students need to write on their own sentence [17].

From the discussion above, the proposed algorithm can produce optimal result for sentences (reference answer and student answer) whose length are almost similar, with tolerance of 3 word difference. With 3 different words, the similarity value >= 0.80 or larger than 80% of similarity value of two compared sentences. The similarity value will drop when the length of the sentences is not similar.

V. CONCLUSION

Based on the experiment, the proposed method that uses term based approach can improve the performance of similarity value computation process between reference answer and student answer. The proposed method is expected to improve the performance of ASAS system by employing simple method and can be used directly to compute similarity value between two sentences. The proposed method can improve the average correlation value on the data set up to 0.56 compared to the other of two weighted methods (Weighted Jaccard Coefficient=0.52 and Weighted Dice Coefficient=0.55). It can also assign similarity value > 0.8 on sentences whose length is almost identical, with tolerance of up to 3 words. Another advantage of the proposed method is its ability to be used directly to compute the similarity value between student answer and reference answer. With such system, it is expected to make it easier in implementing ASAS system for all types of subject matters.

For future research in the area of proposed method, one may investigate a formula to reduce the noise of the unwanted words. One might also add a scheme to understand the

semantic meaning of the compared sentence, by adding thesaurus database. Such addition of database is expected to increase the performance of the proposed method.

REFERENCES

[1] S. Valenti, F. Neri, and A. Cucchiarelli, "An Overview of Current Research on Automated Essay Grading," *J. Inf. Technol. Educ.*, vol. 2, pp. 3–118, 2003.

[2] Z. Feng, "The algorithm analyses and design about the subjective test online basing on the DOM tree," Proc. - Int. Conf. Comput. Sci. Softw. Eng. CSSE 2008, vol. 5, pp. 577–581, 2008.

[3] M. J. Ab Aziz, F. D. Ahmad, A. A. Abdul Ghani, and R. Mahmod, "Automated Marking System for Short Answer examination (AMS-SAE)," 2009 IEEE Symp. Ind. Electron. Appl., no. Isiea, pp. 47–51, Oct. 2009.

[4] A. M. Ben Omran and M. J. Ab Aziz, "Automatic essay grading system for short answers in English language," J. Comput. Sci., vol. 9, no. 10, pp. 1369–1382, 2013.

[5] F. Noorbehbahani and A. a. Kardan, "The automatic assessment of free text answers using a modified BLEU algorithm," Comput. Educ., vol. 56, no. 2, pp. 337–345, Feb. 2011.

[6] F. Rodrigues and L. Araújo, "Automatic assessment of short free text answers," in CSEDU 2012 - Proceedings of the 4th International Conference on Computer Supported Education, 2012, vol. 2, pp. 50–57.

[7] A. Sankar, "Towards an automated system for short-answer assessment using ontology mapping," Int. Arab J. e_technology, vol. 4, no. Guha, p. 1989, 2015.

[8] M. Mohler and R. Mihalcea, "Text-to-text Semantic Similarity for Automatic Short Answer Grading," Proc. 12th Conf. Eur. Chapter Assoc. Comput. Linguist. (EACL '09), no. April, pp. 567–575, 2009.

[9] W. Gomaa and A. Fahmy, "Short Answer Grading Using String Similarity And Corpus-Based Similarity," Int. J. Adv. Comput. Sci. Appl., vol. 3, no. 11, pp. 115–121, 2012.

[10] W. H. Gomaa and A. A. Fahmy, "Automatic scoring for answers to Arabic test questions," Comput. Speech Lang., vol. 28, no. 4, pp. 833–857, 2014.

[11] R. Siddiqi, C. J. Harrison, and R. Siddiqi, "Improving Teaching and Learning through Automated Short-Answer Marking," IEEE Trans. Learn. Technol., vol. 3, no. 3, pp. 237–249, Jul. 2010.

[12] J. Z. Sukkarieh, S. G. Pulman, and N. Raikes, "Auto-marking: using computational linguistics to score short, free text responses," Annu. Conf. Int. Assoc. Educ. Assess. (IAEA), Manchester, UK, pp. 1–15, 2003.

[13] P. Kudi, A. Manekar, K. Daware, T. Dhatrak, and S. Foundation, "Online Examination with Short Text Matching," IEEE Glob. Conf. Wirel. Comput. Netw., pp. 56–60, 2014.

[14] L. Cutrone and M. Chang, "Auto-Assessor: Computerized Assessment System for Marking Student's Short-Answers Automatically," 2011 IEEE Int. Conf. Technol. Educ., pp. 81–88, Jul. 2011.

[15] V. Thada and D. Jaglan, "Comparison of Jaccard, Dice, Cosine Similarity Coefficient To Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm," Int. J. Innov. Eng. Technol., vol. 2, no. 4, pp. 202–205, 2013.

[16] X. Quan, W. Liu, and B. Qiu, "Term weighting schemes for question categorization.," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 5, pp. 1009–21, May 2011.

[17] S. Burrows, I. Gurevych, and B. Stein, The Eras and Trends of Automatic Sh*ort Answer Grading*, vol. 25. 2015.