IJASST International Journal of Applied Sciences and Smart Technologies

Volume 06
Issue 02

p-ISSN 2655-8564, e-ISSN 2685-943, pp. 337-348, 2024

Essay Answer Detection System Uses Cosine Similarity and Similarity Scoring in Sentences

Diah Hidayatul Ula[1], Siti Yuliyanti[1*]

[1]*Informatics Department, Faculty of Engineering,*
Universitas Siliwangi,Indonesia.
[*]*Corresponding Author: sitiyuliyanti@unsil.ac.id*

(Received 12-07-2024; Revised 06-08-2024; Accepted 08-08-2024)

Abstract

Essay exams are often an option to evaluate a person's understanding and interpretation of the material they have studied, rather than simply testing knowledge or understanding through essay exams that still rely on manual methods. The disadvantage of manual assessment is that it is prone to errors due to variations between examiners in providing assessments, coupled with the number of questions that must be assessed which is sometimes quite large, so it takes significant time. So this research carried out the development of an application with an automatic answer correction model with cosine similarity to measure how similar or how far two vectors are in multidimensional space. The result, system's potential in educational contexts was demonstrated by testing it on essay responses. To improve accuracy and usability, future developments could use sophisticated text scoring algorithms and provide more features. This study highlights the importance of automated grading systems for optimizing essay scoring in educational settings while maintaining scalability and reliability.

Keywords: essay exams, automated grading, essay checker, cosine similarity

1 Introduction

Exams with essay questions are frequently used to gauge a student's comprehension and interpretation of the content they have studied. This is a result of teachers' desire to probe more into students' comprehension as opposed to only using multiple-choice exams to assess students' knowledge [1]. Nevertheless, in spite of these benefits, essay exams are still assessed manually. The disadvantage of manual grading is that it is labor-intensive and error-prone due to grade discrepancies among examiners. In addition,

grading a large number of questions might take a considerable amount of time. This is why an application that can automatically correct student answers is an urgent need, expected to help teachers in conducting essay exam grading more quickly and efficiently [2]

This study employed the cosine similarity method in conjunction with a prototype model as a system development approach. A mathematical method called the cosine similarity method can be used to calculate the degree of similarity between two vectors in multidimensional space. Cosine similarity is a tool used to compare how close student answers are to the right key answers while editing essay test responses [3].

In previous research [1], entitled 'Application of Automatic Essay Answer Assessment Using Web-Based Synonym Recognition and Cosine Similarity Methods', an automatic essay answer assessment system was developed using cosine similarity, synonym recognition and filtering methods. Test results show that these systems have a high error rate due to low similarity between student answers and answer keys, typing errors, and missing words in the filtering process. Although it can help in the process of automatically correcting student answers, improvements such as increasing the level of text similarity, creating more dynamic applications, and increasing accuracy testing are needed to improve the performance of automated essay answer grading systems [4]. Research[2], [3], has proven that the use of cosine similarity can produce an accuracy of 89.5%, but there are several types of questions that are different because there are unique words and the answer key does not contain the keywords that are searched for with the correct answer.

Another study [4], et al. entitled "Application of Automatic Essay Exam Grading Using the Cosine Similarity Method" discusses the development of an automatic essay grading system using the Latent Semantic Analysis, Cosine Similarity, and Rabin Karp methods. The research highlights different approaches to automatic essay scoring, focusing on the application of the cosine similarity method to the scoring of essay exams in English. The discussion includes text processing techniques such as stop word removal and stemming, term frequency calculation, word weighting, and similarity calculation using cosine similarity. The results show that the system is effective in overcoming subjectivity in essay grading. The paper also discusses the conversion of similarity scores

into exam scores, the development of an essay auto-grading application, and presents the conclusions and implications of the research [5]. Automated essay scoring is a task implementation of machine learning in NLP, by building money modeling implemented to automate the value of answers to essay questions[5], [6].

Using this method, the system can automatically evaluate students' answers and assign a score based on how close their answers are to the correct answer. This allows teachers to correct essay answers quickly and efficiently without the need for time-consuming manual grading [7], [8]. This cosine similarity method helps overcome essay grading challenges such as examiner inconsistency and a large number of questions. This research applies the model to an answer checking system with advantages such as being able to assess answers quickly and objectively [9], where the assessment model consists of three types, namely multiple choice, true false and essay (description). Descriptive answers are an appropriate method for assessing the results of learning activities, because descriptive answers will involve students' ability to remember and express the ideas they have [10], [11]. The problem of assessment is the description of subjectivity, the assessment between one lecturer and another lecturer may be different or objectivity from different points of view, so it is possible that the lecturer makes mistakes in assessment such as the same student's answer but has a different value.

2 Material and Methods

The stages of this research include collecting datasets, text preprocessing, weighting trams, mapping answers using sosine similarity and scoring, then visualizing test scores which are illustrated in Fig. 1.

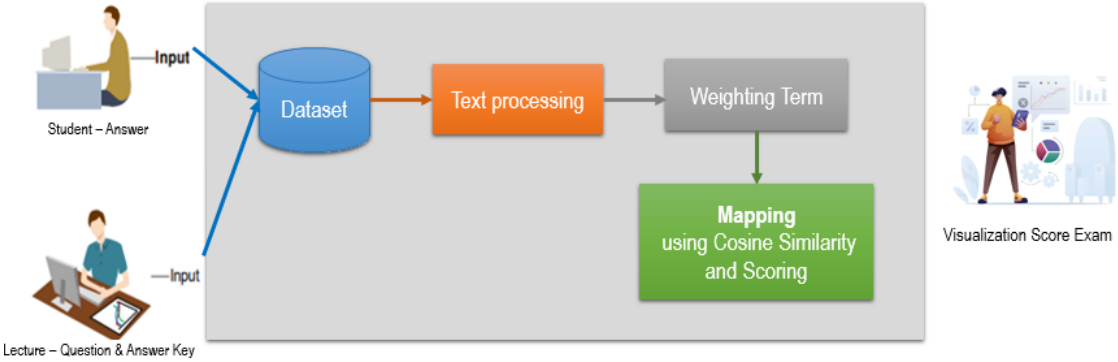


Figure 1. Framework Research

Dataset Collect

Dataset collect using essay assesments and answer key from student. Essay assessments can also be considered assessment tests because they consist of organized questions with answers that provide clarification or additional information. Teachers are still the ones who decide how best to assess their students' talents, usually through essay exams. Because each student's answer is considered objective [12], the assessment process is not actually simple[13] [14], [15]. The essay test is a method or tool that is considered to be highly successful in determining students' academic success as well as their ability to think critically [6].

Text Preprocessing

Text that is to be used as a data source for the next stage can be prepared using text preprocessing [7]. At this point, a number of complex types of actions are possible. They are complicated in nature. The following are the steps that can be performed in text preprocessing:

a. Case Folding

   To make it easier to identify sentences at this point in the process, a text document is converted to lowercase.

b. Tokenization Process

The creation of tokens processing tokens that are present in a collection of data is known as tokenization [5]. In a sentence, symbols based on tabulation, spaces, enter, commas (,), and periods (.) are used to separate each word.

c. Stop-word Removal

At this point, the stop-word is removed as part of the filtering process. Here, a "stop-word" is a term that frequently occurs but has no real significance[4], [16]. The solution is to create a library of stop-words that can be eliminated, like "which, from, in, until, to, and with," which is a mix of the most common word and the stop list [8].

d. Stemming

At this point, affixes—prefixes and suffixes—are eliminated from each document in order to look for commonly used fundamental words. This procedure basically makes use of a database known as the basic-word list [9]. Stemming is a preprocessing part of an information retrieval system which aims to change words in a document into basic words with a certain set of rule [3].

Term Weight

The process of assigning a weight to words in a document according to how often they occur is known as term weighting [17]. This plan can combine Term Weight TF-IDF, which is something that should be taken into account based on a number of papers. In this case, Term Weight - an indicator - must be assigned to each term [10].

a. Terms Frequency

Based on how often a term occurs in a document, a weight is applied [18]. The more occurrences there are, the higher the weight and the higher the match rate (high TF).

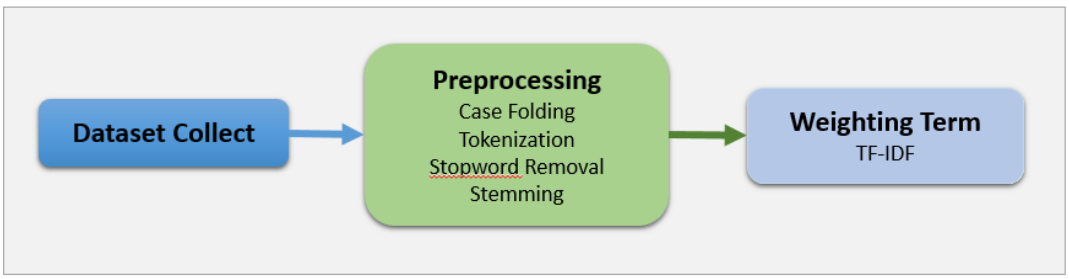


Figure 2. Text Processing

b. Inverse Document Frequency

An estimate derived from the distribution of a word over a set of documents [5], [19]. The IDF shows the relationship between the availability of a word across all documents. The IDF value increases as the number of documents containing the word decreases.

c. TF-IDF

This approach combines two different types of notions for calculating the weight: the inverse frequency of documents containing a word and the frequency of a word occurring in a particular document of the term[17], [19]. The frequency of occurrence of a word in a given document indicates the importance of the word. Importance of the word in the text. The number of documents containing a word indicates how common it is. For the association between a word and a document to have a high weight value, the word must appear frequently in the document and the total frequency of all documents in the document collection that contain low-frequency terms must be high [11]. The formula 1 that can be used for TF-IDF.

Similarity calculation using cosine similarity

In this step, the similarity between the students' essay answers and the instructor's key answers is calculated using the following formula 2 and 3.

$$Similarity = \cos(\theta) = \frac{A.B}{|A||B|} \quad (1)$$

At this stage, only two responses are analyzed, so the degree of similarity between the two responses is the output, which is then translated into student scores[6], [20].

Conversion of Similarity Score to Essay Exam Score

The similarity scores generated earlier are converted into answer scores for the of the student's essay exam[5], [16]. These values are based on the human judgment version of the value range, which is a reference from Fuat's research in 2010 following in Table 1.

Table 1. Range of Score

Rentang Nilai "Essai Checker"	
Similarity	**Grade**
90-100	**A**
80-89	**B**
70-79	**C**
60-69	**D**
<59	**E**

3 Results and Discussions

The Fig. 3 shows a web-based tool called "Essay Checker". Its purpose is to facilitate the grading of student essays using the cosine similarity approach. Users of the system, such as instructors or professors, can upload two Excel files: "Excel Kunci Jawaban" and "Excel Jawaban Siswa", which contain the solution keys to the students' answers.

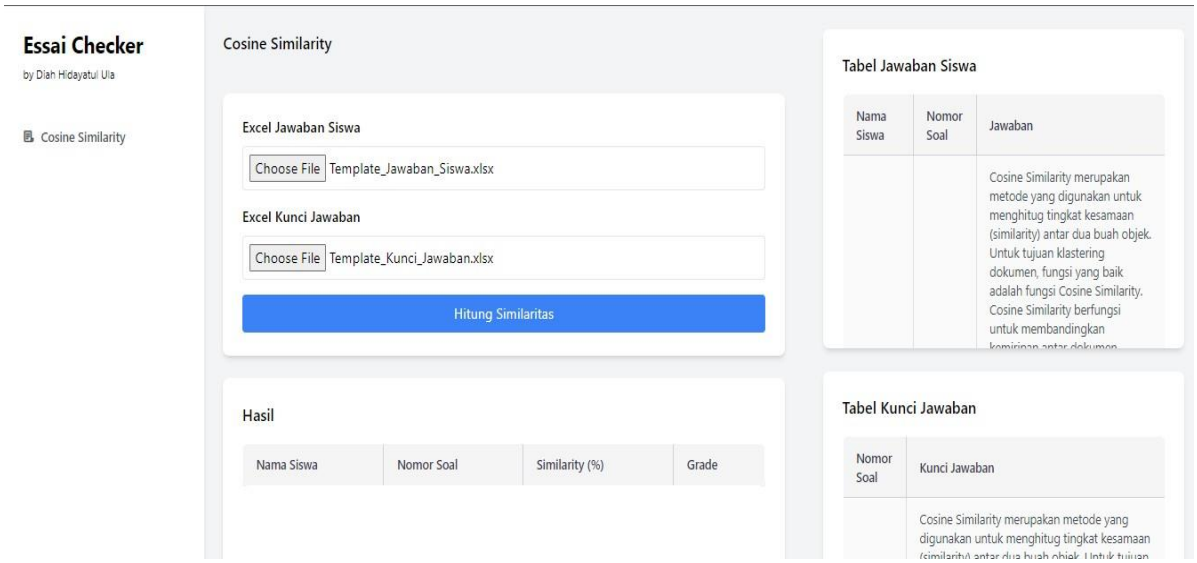


Figure 3. Website "Essai Checker"

The Fig. 4 shows two Excel files to be uploaded into the “Essai Checker” system. The first file, "Template_Jawaban_Siswa", contains the essay answers of several students. This file has three columns: "NAMA_SISWA" which contains the student's name, "NO_SOAL" which contains the question number, and "JAWABAN" which contains the student's answer text for each question. For example, there is a student named "DIAH" who provided answers to question number 1 on cosine similarity, as well as a student named "JISOO" who provided answers to several other questions.

The second file, "Template_Kunci_Jawaban", contains the correct answer key for each question. This file has two columns: "NO_SOAL", which contains the question number, and "JAWABAN", which contains the text of the answer that is considered correct for each question. For example, the answer to question number 1 is a description of cosine similarity and how it works, while the answer to question number 2 is an explanation of the definition of a system.

These two files are uploaded into the “Essai Checker” system to calculate the percentage of similarity between the student's answers and the answer key using the cosine similarity method. The system will process the data from both files and produce a result table showing the degree of similarity and grade for each student answer based on its compatibility with the provided answer key.

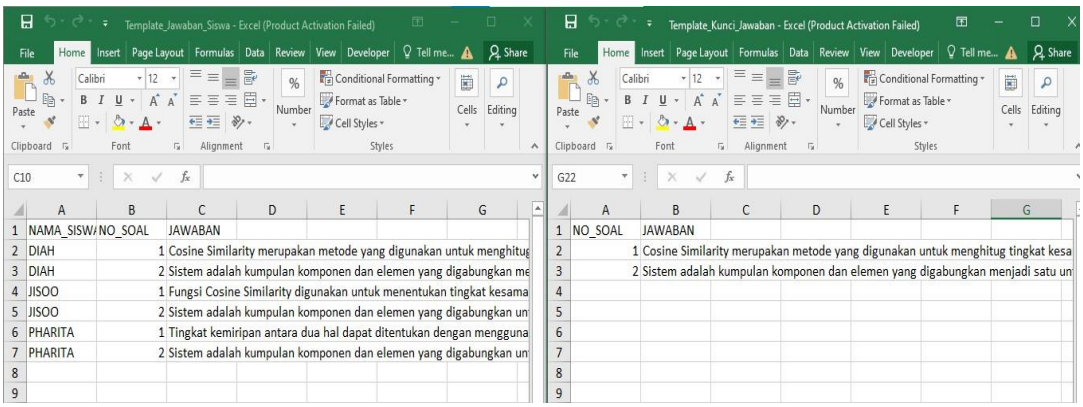
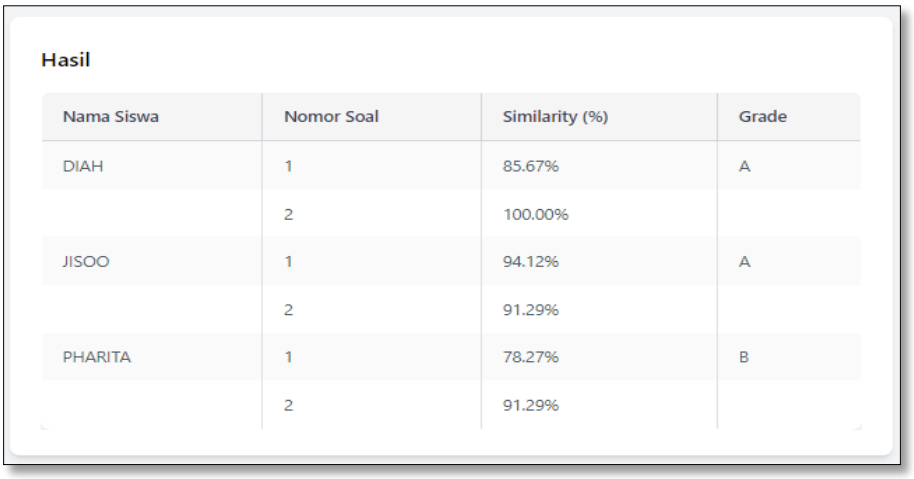


Figure 4. File Uploaded



Hasil

Nama Siswa	Nomor Soal	Similarity (%)	Grade
DIAH	1	85.67%	A
	2	100.00%	
JISOO	1	94.12%	A
	2	91.29%	
PHARITA	1	78.27%	B
	2	91.29%	

Figure 5. Results for the calculation

After both files are uploaded, the user can press the "Hitung Similaritas" button to start the process of calculating the similarity between the student's answer and the answer key.

The results of this calculation are then displayed in several tables as shown in Fig. 5. “Jawaban Siswa table” displays information about the student's name, the question number and the answer given. “Kunci Jawaban” table displays the question number and the correct answer according to the answer key. In addition, the "Hasil" table displays the percentage of similarity between student answers and answer keys in the form of numbers. In addition, this system also provides a grade based on the degree of similarity.

4 Conclusions

In this research, the authors successfully implemented the cosine similarity method on a web-based automatic assessment system of essai answers based on climate matching. Overall, the "Essay Checker" system proved its potential as an effective tool in the essay assessment process in an educational environment. Further development could include integration with more advanced text scoring algorithms and the addition of new features to improve the accuracy and functionality of the system.underway.

Acknowledgements

We extend our deepest gratitude to the professor in charge of the Natural Language Processing course for their guidance and support throughout this research. We also thank all parties who have contributed to this research.

References

[1] A. Salim, K. Rijal, and B. Hendrik, "Studi Literatur Sistem Penilaian Esai Otomatis Pada E-Learning Dengan Algoritma Winnowing," *J. Sist. Inf. dan Ilmu Komput.*, vol. 1, no. 3, pp. 163–172, 2023, [Online]. Available: https://doi.org/10.59581/jusiik-widyakarya.v1i3.1227

[2] A. C. Herlingga, I. P. E. Prismana, D. R. Prehanto, and D. A. Dermawan, "Algoritma Stemming Nazief & Adriani dengan Metode Cosine Similarity untuk Chatbot Telegram Terintegrasi dengan E-layanan," *J. Informatics Comput. Sci.*, vol. 2, no. 01, pp. 19–26, 2020, doi: 10.26740/jinacs.v2n01.p19-26.

[3] R. R. Et.al, "The Similarity of Essay Examination Results using Preprocessing Text Mining with Cosine Similarity and Nazief-Adriani Algorithms," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 3, pp. 1415–1422, 2021, doi: 10.17762/turcomat.v12i3.938.

[4] R. I. Pratama, Munir, and R. Megasari, "Detector Similarity Answers Between Students on Essay Digital Exam System," *Proc. 7th Math. Sci. Comput. Sci. Educ. Int. Semin. MSCEIS 2019*, 2020, doi: 10.4108/eai.12-10-2019.2296348.

[5] T. Wahyuningsih *et al.*, "Text Mining an Automatic Short Answer Grading (ASAG), Comparison of Three Methods of Cosine Similarity, Jaccard Similarity and Dice's Coefficient," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 2, pp. 343–348, 2021, doi: 10.17762/turcomat.v12i3.938.

[6] W. A. S. S L B Ginting, Y R Ginting, Sutomo, "Aplikasi Deteksi Kemiripan Kata Menggunakan Algoritma Rabin-Karp," *J. Teknol. dan Inf.*, vol. 12, no. 2, pp. 162–175, 2022, doi: 10.34010/jati.v12i2.

[7] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[8] F. Al-Turjman, H. Zahmatkesh, and L. Mostarda, "Quantifying uncertainty in internet of medical things and big-data services using intelligence and deep learning," *IEEE Access*, vol. 7, pp. 115749–115759, 2019, doi: 10.1109/ACCESS.2019.2931637.

[9] J. R. Saura, B. R. Herraez, and A. Reyes-Menendez, "Comparing a traditional approach for financial brand communication analysis with a big data analytics technique," *IEEE Access*, vol. 7, pp. 37100–37108, 2019, doi: 10.1109/ACCESS.2019.2905301.

[10] D. Nallaperuma *et al.*, "Online Incremental Machine Learning Platform for Big Data-Driven Smart Traffic Management," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4679–4690, 2019, doi: 10.1109/TITS.2019.2924883.

[11] M. M. Fawzy, A. S. Elsharkawy, Y. A. Khalifa, and A. A. hassan, "Contractor selection by using multi-criteria decision-making for Egyptian road maintenance," *Int. J. Syst. Assur. Eng. Manag.*, 2024, doi: 10.1007/s13198-024-02249-3.

[12] S. Yuliyanti and Rizky, "Implementasi Algoritma Rabin Karp Untuk Mendeteksi Kemiripan Dokumen Stmik Bandung," *J. Bangkit Indones.*, vol. 10, no. 02, p. 1, 2020, doi: 10.52771/bangkitindonesia.v10i02.124.

[13] S. Legianto, "Implementasi Text Mining Untuk Mendeteksi Hate Speech Pada Twitter," p. 60, 2019.

[14] R. A. Raharjo, I. M. G. Sunarya, and D. G. H. Divayana, "Perbandingan Metode Naïve Bayes Classifier Dan Support Vector Machine Pada Kasus Analisis Sentimen Terhadap Data Vaksin Covid-19 Di Twitter," *Elkom J. Elektron. dan Komput.*, vol. 15, no. 2, pp. 456–464, 2022, doi: 10.51903/elkom.v15i2.918.

[15] F. Muhammad, N. M. Maghfur, and A. Voutama, "Sentiment Analysis Dataset on COVID-19 Variant News," *Sci. J. Inromation Syst. Informatics*, vol. 4, no. 1, pp. 382–391, 2022.

[16] L. R. Setiawan and D. Nasien, “Cosine Similarity for Essay Answer Detection,” *J. Appl. Bus. Technol.*, vol. 1, no. 1, pp. 36–40, 2020, doi: 10.35145/jabt.v1i1.21.

[17] F. Setio Pribadi, U. Hasanah, and D. Nur Sa, “Automatic Short Answer Scoring (ASAS) Using String-based Similarity and Query Expansion,” pp. 678–684.

[18] B. L. Qasthari, E. Susanti, and M. Sholeh, “Classification Of Lung and Colon Cancer Histopathological Images Using Convolutional Neural Network (CNN) Method An A Pre-Trained Models,” *Int. J. Appl. Sci. Smart Technol.*, vol. 5, no. 1, pp. 133–142, 2023, doi: 10.24071/ijasst.v5i1.6325.

[19] E. L. Amalia, A. J. Jumadi, I. A. Mashudi, and D. W. Wibowo, “Analisis Metode Cosine Similarity Pada Aplikasi Ujian Online Otomatis (Studi Kasus JTI POLINEMA),” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 2, pp. 343–348, 2021, doi: 10.25126/jtiik.2021824356.

[20] N. Vendyansyah and Y. A. Pranoto, “Perancangan dan Pembuatan Aplikasi untuk Mendeteksi Kemiripan Jawaban Menggunakan Cosine Similarity,” *J. Tek. (Jurnal Fak. Tek. Univ. Islam Lamongan)*, vol. 13, no. 1, pp. 23–28, 2021.