

Development of Moodle-based Plugin for Automated Essay-Type Grading

Fajar Pradana^{*1}, Welly Purnomo², Atthoriq Adillah³, Buce Trias Hanggara⁴

^{1,2,3,4} University of Brawijaya, Malang

¹fajar.p@ub.ac.id, ²wepe@ub.ac.id, ³thoriqadillah59@gmail.com,

⁴buce_trias@ub.ac.id

*Corresponding Author

Received 30 August 2023; accepted 26 Maret 2024

Abstract. The essay is an exam requiring a more profound understanding of answering and evaluating the answers. However, if the number of questions and participants increases, this will result in a decrease in the quality of the lecturer's assessment. This prompted the development of a software based on the Moodle plugin named Essay Similarity to assess essay answers based on the similarity between the two documents, namely the answers given and the answer key provided. Moodle was chosen because, after the Covid-19 pandemic, many sectors have shifted to working remotely, including the education sector. This resulted in an LMS like Moodle experiencing an increase in users. As of 2020 yesterday, Moodle users have exceeded 190 million users on more than 145,000 websites. In developing this plugin, the method used is the waterfall method using the PHP programming language. The algorithm used to find similarities between the two documents is cosine similarity. Testing the level of similarity between manual and automatic grading was carried out on four models of essay questions. Based on the test results, the average similarity between automatic grading compared to manual grading is 45.44%.

Keywords: moodle, cosine similarity, essay, exam, plugin

1 Introduction

Since the outbreak of COVID-19, much work that was done initially face-to-face has now become online. This resulted in the increased use of technology, such as the Internet, worldwide, especially in Indonesia. The Central Bureau of Statistics reports that 78.18% of households in Indonesia have used the Internet. This figure is an increase of 4.43% from last year [1]. One of the uses of information system technology is the Learning Management System (LMS). Various Indonesian educational institutions, such as universities, use LMS as a medium for distance learning. One example of an LMS is Moodle. Moodle is a web-based LMS developed in PHP. By using Moodle, distance and face-to-face learning can be helped by features such as managing assignments, exams, and materials. Furthermore, the functionality and capabilities of Moodle can be expanded through the integration of plugins created by established communities, thereby enhancing the overall quality and features of the Moodle platform.

Even in this completely automated era, Moodle still does not automatically assess their essay exam answers. By default, teachers must manually assign grades to the essay answers. This is because the essay-type exam is a type of exam that requires a more profound understanding of answering and evaluating these answers. However, suppose

the number of questions and participants is increasing. In that case, this will decrease the quality of the assessment from a lecturer, which will make the assessment given sometimes inconsistent.

When viewed from the selection of existing plugins, at the time of writing this research, only two plugins automatically graded essay exam answers. Even so, the functionality of the plugin is still relatively limited. Each of these plugins has a different technique for automatically grading essay answers. And each also has weaknesses in its application. The first plugin is a plugin called "Essay (auto-grade)." In a plugin called "Essay (auto-grade)," an automatic grading process is carried out by checking whether a keyword is available in the answers given, not guessing whether the answers given are similar to the existing answer keys. In other words, the "Essay (auto-grade)" plugin uses keyword techniques that must be present in student answers. Even though this technique is legitimate to use, and sometimes lecturers also evaluate the answers to essay questions manually utilizing this technique to save time and effort, this approach cannot know the context of the answers given. This is because a solution may not only contain keywords so that the answer is correct or reasonable but also words that support and form a context are needed to consider the answer correct or reasonable. The second plugin is a plugin called "Pattern match." In a plugin called "Pattern match," an automatic assessment is carried out by analyzing the word order patterns of the answers given. With "Pattern Match," the plugin can find out whether the word order of the answers given is inverted or not. To get these results, the question maker must manually make a series of expressions if a word has a different order or is synonymous with another word. This is a weakness of the "Pattern Match" plugin because of a learning curve or a pattern that the question maker must learn. Similar to the previous plugin, this "Pattern Match" plugin cannot know the context of the answers given by students.

Based on the previously mentioned problems, a solution was devised by developing a Moodle plugin that can automatically grade essay exam answers with more flexible boundaries than previous plugins called Essay Similarity. In its implementation, the cosine similarity algorithm checks the similarity between the answer keys and the answers given. Cosine similarity is an algorithm used to measure similarity that can be used to compare the similarity of a document with other documents [2]. Cosine similarity has been widely used in various applications, such as the classification of journals based on their abstracts [3], the type of thesis documents [4], and the equality of essay exam results [5].

Based on the description and implementation of the cosine similarity above, the cosine similarity algorithm is considered suitable for solving the previously described problems. The method used in this development is the waterfall method. The waterfall is a development method divided into requirements specification, software design, implementation, testing, and so on, carried out linearly and sequentially [6]. The waterfall method is used because the way is linear, and user needs are defined at the start and do not change frequently. This allows researchers to carry out one process at a time and minimize the repetition of the process due to limited time. As for what later is, a general requirement of this plugin is to be able to provide an automatic grading of the answers to essay exam questions by comparing the similarities between the answer keys and the solutions are given. This explanation makes the waterfall method suitable for use in this study [7].

2 Literature Review

In the research conducted by Muhammad, Arwani, and Rahayudi [8], the Moodle plugin is used to send notifications to users using Telegram BOT. This research is used

as a reference because the development and programming language used is the same, namely the development of Moodle plugins using the PHP programming language. This is what makes this research suitable as a reference.

Fanani, Aknuranda, and Purnomo successfully conducted research to develop a Goat Farming Financial Management Information System using Waterfall [9]. Due to the similarity of the method used, this research is suitable as a reference for conducting this research.

In addition to the research that has been done, the literature review in this research is also taken from plugins that are similar to those that will be developed. The plugins that are similar to the plugins that will be developed in this research are the plugins named "Essay (auto-grade)" and "Pattern Match." The two plugins were taken as references for the development of plugins in this study because of the same type of plugin, namely the question type plugin.

As for measuring the similarity between two documents, we can use Natural Language Processing (NLP). NLP is a computer engineering theory for analyzing and representing texts that occur naturally at one or more levels of linguistic analysis like humans process language, which is then used for various tasks or applications [10]. NLP is commonly used when we want to transform information in text into structured data that computers can understand [11]. Many applications utilize NLP, one of which is text similarity. Two well-known algorithms can be used to find similarities between two documents, namely cosine similarity and Euclidean distance

2.1 Cosine Similarity

Cosine similarity is part of Natural Language Processing (NLP). Cosine similarity is an algorithm used to measure similarity, which can be used to compare the similarity of a document with other documents. The cosine similarity algorithm can calculate how similar two documents are regarding their subject [12]. One of the reasons why cosine similarity has been chosen is because of the algorithm's efficiency in evaluating the similarity between two documents [13]. The cosine equation used in this literature was:

$$\text{cosine}(d1, d2) = \frac{(d1 \cdot d2)}{\|d1\| \|d2\|} \quad (2.1)$$

whereas $(d1, d2)$ represents the dot product of the two documents, meanwhile $\|d\|$ represents the length of a document.

In addition to cosine similarity, there are methods for finding similarities between two documents, namely the Euclidean distance. Euclidean distance is a geometric measure used to measure the distance between two vectors so that from this distance, you can see the similarity between the two documents [14].

This research will later use cosine similarity as the algorithm of choice because cosine similarity is proven more efficient, especially in case studies of automatic grading of essay questions compared to Euclidean distance [12]. In addition, cosine similarity is better to use than Euclidean distance because cosine similarity is not affected by variables with tremendous upper and lower limit values that affect the similarity results. Additionally, cosine similarity considers the angle of similarity between two documents, unlike Euclidean distance. Therefore, based on these justifications, cosine similarity was selected as the preferred approach for addressing the issues raised in this study, instead of utilizing Euclidean distance.

2.2 Waterfall Model

The waterfall is a development method divided into requirements specification,

software design, implementation, testing, and so on, carried out linearly and sequentially [6]. The waterfall method is suitable when the user's needs are defined initially and do not change frequently [7]. This is because the process in the waterfall method is linear, meaning that a process will only continue if the previous operation has been completed.

3. Methodology

This research is an implementation type, with the main product being the Moodle plugin. The processes carried out in this study were based on the waterfall method, described previously with several adjustments, namely starting with a literature study and ending with drawing conclusions and suggestions, as shown in Fig 1. This research started with requirement engineering, the design phase, implementation in code from the design-build, testing, automated scoring testing, and ended with the conclusion stage.

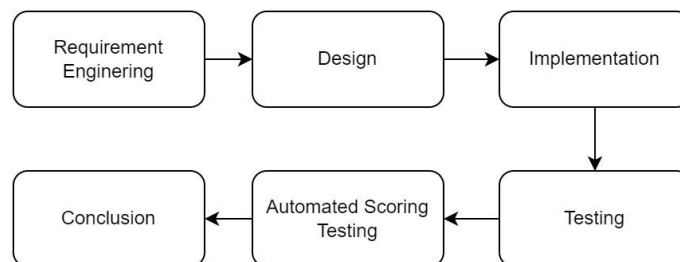


Figure 1. Methodology

3.1 Requirement Engineering

Requirements analysis is carried out to determine what the system needs. In developing the Essay Similarity plugin, the required research was done using direct observation techniques (participant observer). The things that will be observed are plugins that are similar to the Essay Similarity plugin, as well as observing how the cosine similarity algorithm works itself. This needs analysis is essential because the needs analysis results will be used to develop the Essay Similarity plugin. In terms of gathering the functional and non-functional systems, observation and interviews with some stakeholders such as lecturers and students were conducted.

3.2 Design

At this stage, the needs analysis is carried out from the needs analysis that has been done before. The description of these requirements is like separating between functional and non-functional requirements. After that, modeling was carried out using the Unified Modeling Language (UML) based on the needs analysis that had been done before. As the modeling used in this study is the use case diagram, use case scenario, system architecture, class diagram, physical data model, and sequence diagram. But, this paper would only describe the use case diagram and the system architecture used.

3.3 Implementation

The implementation phase includes development in the form of writing plugin code from start to finish using the PHP programming language based on the designs that have been made before. The implementation also includes some functionality description and pseudo-code for that step. The interface of the system will also be shown in this section.

3.4 System Testing

The system testing phase is carried out to look for possible bugs in the system. Often, developers miss the implementation stage and accidentally distribute software with bugs. Therefore, this testing phase aims to minimize bugs in the software. Moreover, this software is a plugin that the general public will use. The method used in system testing is black box testing.

3.5 Testing of Automatic Grading Results

The testing phase of the results of the automatic grading is used to measure the degree of similarity between the manual assessment and the automatic grading. This is done to find out how close the value generated by the automatic grading is to the manual evaluation by the lecturer. As for the method in this test, it is carried out using the Mann-Whitney U test and making comparisons between manual appraisals and automatic gradings. This test involved 25 students as the object of the test.

3.6 Conclusions

Conclusions and Suggestions is the last stage of this research, where conclusions and suggestions are drawn from this research. The conclusion is a statement from the ideas reached at the end of the research related to the questions asked in the previous problem formulation. While suggestions are various suggestions that can be used to improve the quality of Essay Similarity plugin development

4 Result and Discussion

4.1 Requirement Analysis

Based on how cosine similarity works. To look for similarities between documents, having two or more documents is necessary. In the context of this plugin, it will use two documents. The first document is the lecturer's answer key, and the second is the student's answer. So from here, the functional requirements based on the actors described earlier are shown in Fig 2. Lecturer actors can create, edit, delete, and answer questions. Questions in this context help specify how the first document was created. Then for student actors, the useful thing to do is answer questions. Answering questions is helpful as a second document is needed to assess the similarity of the two documents. Meanwhile, the NLP component has a function to calculate the similarity between the two documents.

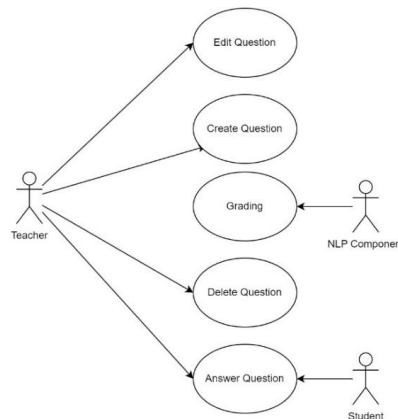


Figure 2. Use-case Diagram

4.2 Design

Fig 3. shows the system architecture that will be used as a reference at the implementation stage. The components below are interconnected. Starting from making a quiz, the actor will choose the type of question in the Essay Similarity plugin. From there, the question or questions will be asked. Based on the questions created earlier, it will have related responses or answers, and these answers will also be related to the plugin because these answers answer questions created by the Essay Similarity plugin type of question. After answering the questions, the similarity will be compared using the Natural Processing Language (NLP) component using the cosine similarity algorithm and the pre-processing process. The result of this similarity will be the automatic grading or value of the essay answer.

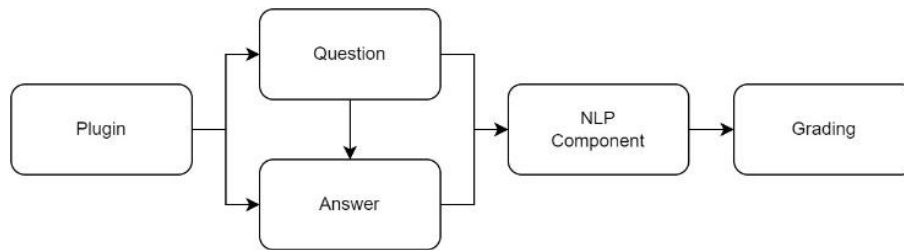


Figure 3. Arcitecture Model

4.3 Implementation

The implementation phase includes development in the form of writing plugin code from start to finish using the PHP programming language based on the system design that was created previously. The following is the result of the implementation that has been carried out.

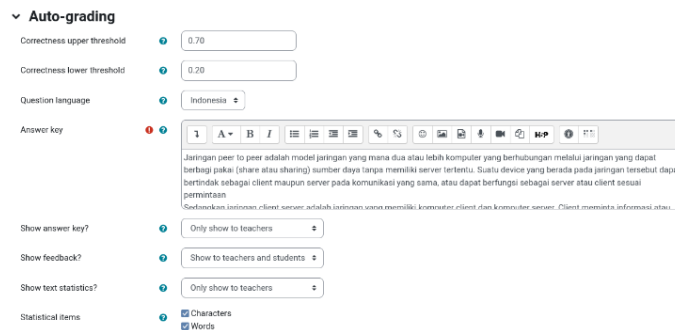


Figure 4. Grading User Interface

Fig 4 and Fig 5 shows the grading functionality; actors can edit questions they have created before. After that, the changes will be saved in the database. In this functionality, actors can answer questions that have been asked before. In answering the questions, there are two views. The first view is to answer the question accompanied by a file, the second is not, and only the text part will be calculated for similarity.

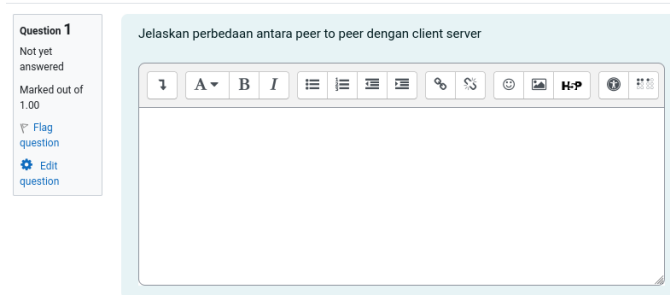


Figure 5. Exam User Interface

4.4 Testing

The tests carried out in this study were about the similarity between manual assessments by a lecturer and automatic gradings using the Essay Similarity plugin. The following are the questions, along with the answer keys, that are used as an automated grading test. Four essay questions are the test sample, so a comparison test is carried out between the automatic grading and manual assessment results to determine whether there is a significant difference between the two assessment methods. Fig 6,7,8,9 show the graph of the automatic and manual assessment results of the four questions. For all the figures, the blue graph indicates the score rated by manual assessment, and the red line shows the result from the automated grading test. In Fig 10 we can also see one of the example of automated grading, which shown score of 0.78 and manually graded set to 0.95.

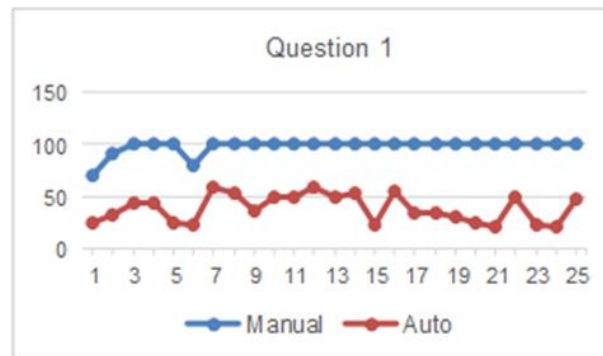


Figure 6. Question 1

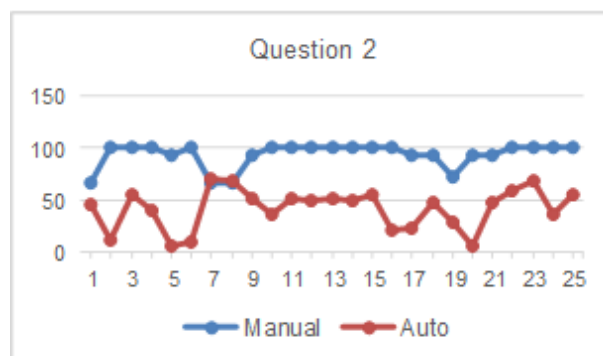


Figure 7. Question 2

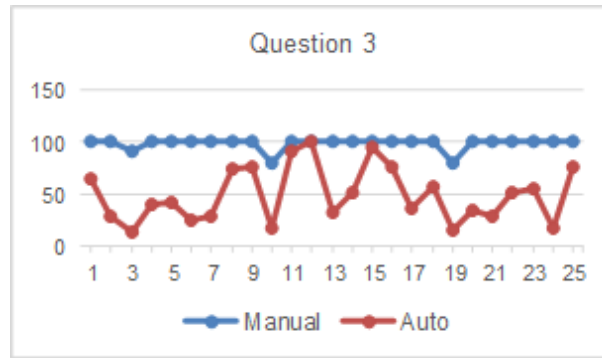


Figure 8. Question 3

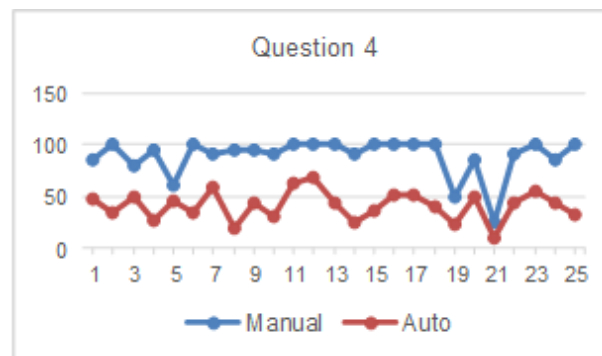


Figure 9. Question 4

Text statistics

- Characters: 1,008
- Words: 141
- Sentences
- Paragraphs
- Unique words
- Long words
- Characters per sentence
- Words per sentence
- Long words per sentence
- Sentences per paragraph
- Lexical density
- Fog index

Feedback

The similarity between answer and the answer key is 0.78. This is automatic grading. The grade may be overridden by the teacher or accepted as is.

Grading

27 Nov 2022 at 14:35

The grade for this essay was manually set to 0.95.

Comment:

Make comment or override mark

Response history

Step	Time	Action	State	Marks
1	27/11/22, 14:25:21	Started	Not yet answered	
2	27/11/22, 14:35:25	Saved: Jaringan peer to peer adalah jenis jaringan yang saling menghubungkan komputer satu dengan yang lain tanpa perlu bantuan switch ataupun hub. Pada jaringan ini tidak mengenal istilah server maupun client, karena semua kedudukan komputer sama dan sejar. Artinya tidak ada komputer yang memiliki peran khusus pada jaringan tersebut, sehingga semua pihak dapat berperan sebagai client ataupun server. Sedangkan client server	Answer saved	

Figure 10. Example of Automated Grading and Manual Grading

To carry out a comparative test, from all of the questions given, it is first necessary to carry out a normality test on all data to determine the proper test method. If the P value <0.05, then the data is not normally distributed, and vice versa. Table I shows the results of the normality test.

Grading	P Value
Manual	0.0000
Automatic	0.095494

Since the P value of one of the data points is less than 0.05, it indicates that the data does not follow a normal distribution. Consequently, the appropriate statistical approach for conducting a comparative test would be the Mann-Whitney U test. This test involves formulating a hypothesis in which the comparative analysis will be performed, considering the non-normal distribution of the data. In this test, a hypothesis will be drawn where:

H0: There is no significant difference between manual and automatic grading

H1: There is a significant difference between manual and automatic grading

In the event that the obtained P value from the Mann-Whitney U test exceeds 0.05, the null hypothesis (H0) is accepted, while the alternative hypothesis (H1) is rejected. Conversely, if the obtained P value is less than 0.05, the null hypothesis (H0) is rejected, and the alternative hypothesis (H1) is accepted. The outcomes of the Mann-Whitney U test are presented in Table 2, providing a summary of the results.

Table 2. Result Of Mann-Whitney U Test

Question	Z Score	P Score
1	6.05369	0
2	5.87906	0
3	5.67533	0
4	5.4619	0

Based on the Mann-Whitney U test above, it can be stated that all questions have a P value <0.05 , which means that there is a significant difference between manual and automatic gradings. This happens because many synonyms and languages are mixed in the students' answers, resulting in many unique words so that cosine similarity cannot detect the similarity between the two documents.

As for making a comparison between manual and automatic grading, the average comparison between questions 1, 2, 3, and 4, respectively, is 39.68%, 45.91%, 49.27%, and 46.92%, and the average comparison between automatic grading and manual as a whole is 45.44%

Acknowledgments. This research is partially supported by the Faculty of Computer Science, Universitas Brawijaya.

References

- 1 Dwi Hadya Jayani, "Penetrasi Internet Indonesia Meningkatkan saat Pandemi Covid-19," <https://databoks.katadata.co.id/datapublish/2021/10/06/penetrasi-internet-indonesia-meningkat-saat-pandemi-covid-19>, 2021.
- 2 J. Han, *Data mining: concepts and techniques*. Elsevier, 2012. doi: 10.1016/C2009-0-61819-5.
- 3 M. Habibi and P. W. Cahyo, "Journal Classification Based on Abstract Using Cosine Similarity and Support Vector Machine," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 4, no. 3, p. 48, Feb. 2020, doi: 10.14421/jiska.2020.43-06.
- 4 R. T. Wahyuni, D. Prastiyanto, and D. E. Suprpto, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi," 2017.
- 5 R. Rosnelly, "The Similarity of Essay Examination Results using Preprocessing Text Mining with Cosine Similarity and Nazief-Adriani Algorithms," *Turkish*

- Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 3, pp. 1415–1422, Apr. 2021, doi: 10.17762/turcomat.v12i3.938.
- 6 I. Sommerville, *Software Engineering*, 9th ed. Pearson, 2011.
 - 7 H. K. Aroral, “Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis,” *International Journal of Applied Business and Management Studies*, vol. 6, no. 1, 2021.
 - 8 R. Muhammad, B. Rahayudi, and I. Arwani, “Implementasi Plugin Notifikasi Sebagai Media Integrasi Antara E-Learning Moodle dengan BOT Telegram (Studi Kasus: Bimbingan Belajar The Second School),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2020.
 - 9 R. I. Fanani, I. Aknuranda, and Y. T. Mursityo, “Pengembangan Sistem Informasi Manajemen Keuangan Peternakan Kambing (Studi Kasus: Yoga’s Farm Kabupaten Tulungagung),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2019.
 - 10 E. D. Liddy, “Natural Language Processing,” 2001. [Online]. Available: <https://surface.syr.edu/istpub>
 - 11 C. Friedman, T. C. Rindfleisch, and M. Corn, “Natural language processing: State of the art and prospects for significant progress, a workshop sponsored by the National Library of Medicine,” *J Biomed Inform*, vol. 46, no. 5, pp. 765–773, Oct. 2013, doi: 10.1016/j.jbi.2013.06.004.
 - 12 O. E. Oduntan and I. Adeyanju, “A Comparative Analysis of Euclidean Distance and Cosine Similarity Measure for Automated Essay-Type Grading,” *Journal of Engineering and Applied Sciences*, 2018.
 - 13 G. Salton, A. Wong, and C. S. Yang, “A vector space model for automatic indexing,” *Commun ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975, doi: 10.1145/361219.361220.
 - 14 A. Heidarian and M. J. Dinneen, “A Hybrid Geometric Approach for Measuring Similarity Level Among Documents and Document Clustering,” in *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, IEEE, Mar. 2016, pp. 142–151. doi: 10.1109/BigDataService.2016.14.