



Comparison of Similarity Methods on New Student Admission Chatbots Using Retrieval-Based Concepts

Arif Mudi Priyatno^{1*}, M. Riko Anshori Prasetya², Putri Cholidhazia³, Resy Kumala Sari⁴

¹Department of Digital Business, University of Pahlawan Tuanku Tambusai, Riau, Indonesia

²Department of Information Systems, University of Sari Mulia, Kalimantan Selatan, Indonesia

³Department of Informatic Engineering, University of Al-Azhar Indonesia, Indonesia

⁴Department Industrial Engineering and Management, Yuan Ze University, Taiwan

Article Info

Keywords:

Similarity Methods
New Student Admission
Chatbot
Retrieval-Based Concepts
Cosine
Dice

ABSTRACT (10 PT)

A college's students are an essential component. The college always opens registration for new students each year. Every year, more than 1,000 prospective new students register. Because of this, the new student admissions committee is constantly overwhelmed when responding to campus-related questions. As a result, developing a chatbot to assist new students is necessary. The best similarity method is needed for the development of a chatbot using a retrieval-model approach. The New Student Admission Chatbot and the Similarity Method are compared in this study using the Retrieval-Based Concept. The cosine, Jaccard, dice, euclidean, Manhattan, Canberra, and Chebyshev similarity methods are compared. In the context of Universitas Pahlawan Tuanku Tambusai, the data used are information about new students as well as accreditation for study program. There are 41 pieces of information used. Labels and information make up data. According to the test results, the dice and cosine similarity methods are the most effective. On all tested thresholds, dice and cosine similarity achieved an f1-score above 80%. Recall produces extremely optimal results, including 100%. Over 75% of the time, good results are reliably achieved. This demonstrates that the retrieval-model concept can be applied to the creation of chatbots using the dice or cosine similarity method.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Arif Mudi Priyatno
Department of Digital Business
University of Pahlawan Tuanku Tambusai
Jl. Tuanku Tambusai No. 23, Bangkinang, Riau, Indonesia 28412
Email: arifmudi@universitaspahlawan.ac.id

1. INTRODUCTION (10 PT)

The university conducts new student admissions (PMB) every year. As a result, the university has an admissions division that handles a variety of new student admissions-related issues. Additionally, the division responds to inquiries from prospective new students. Daily inquiries come from the media that is provided. Accreditation, entry pathways, scholarships, and other topics are frequently asked of prospective

students. On weekdays and during business hours, the PMB Division only responds to inquiries. As a result, the PMB division was unable to adequately respond to the existing questions [1]. Prospective students' access to information becomes sluggish as a result. Making a list of frequently asked questions and their answers, also known as Frequently Asked Questions (FAQ), is one option. Questions that are not addressed in the FAQ should only be addressed by the PMB Division. When there are a lot of questions in the FAQ, issues will arise. Because the entered keywords appear in numerous questions, prospective students will have difficulty locating the relevant data. Because of this, prospective students must enter the keyword "Unique" in order to obtain the necessary information [2].

Technology aids in the resolution of life's issues, including the process of accepting new university students. Technology can perform certain tasks like humans. Artificial Intelligence (AI) technology is the technology. A chatbot is an artificial intelligence technology that will help the PMB division respond to inquiries from prospective students. Chatbot is an innovation intended to have the option to convey among people and PCs through instant messages. In a number of instances, chatbots have been widely implemented. Brazilian Virtual School [3], online commerce [4][5], online courses [6], Covid-19 public health [7], and others are applications of chatbots. For this situation, the chatbot interfaces correspondence between imminent understudies and the PMB division in regards to new understudy data.

Rule-based, retrieval-based, and generative-based approaches are typically used to create chatbots [8][9]. Rule-based is a regular methodology procedure by matching in light of the standards that have been given. Generative-based is a method that takes into account the probabilities of all words and is based on a neural network. An approach technique based on the idea of retrieving information is called retrieval-based. The relevant results are returned after retrieval-based calculates the similarity (similarity) of the keywords. Cosine similarity, dice similarity, jaccard similarity, euclidean similarity, manhattan distance, canberra distance, and chebyshev distance are all methods of measuring similarity (data similarity distance).

In this paper, we comparison of similarity methods on new student admission chatbots using retrieval-based concepts. The data used pertain to the Hero Tuanku Tambusai University's acceptance of new students. The term frequency-inverse document frequency (TF-IDF) method was used to weight the data. Using a variety of similarity methods, the level of similarity is calculated for the weighted data. using the f-measure, precision, and recall to evaluate performance and success. The new student chatbot at Universitas Pahlawan Tuanku tambusai is expected to be implemented using the comparison of this similarity method.

2. RESEARCH METHOD

There are a number of stages in the research methodology for comparing similarity methods for chatbots. The stages are pre-processing, weighting, similarity, performance measurement, and the formation of question data and answers. The stage of the research that was done is shown in Figure 1.

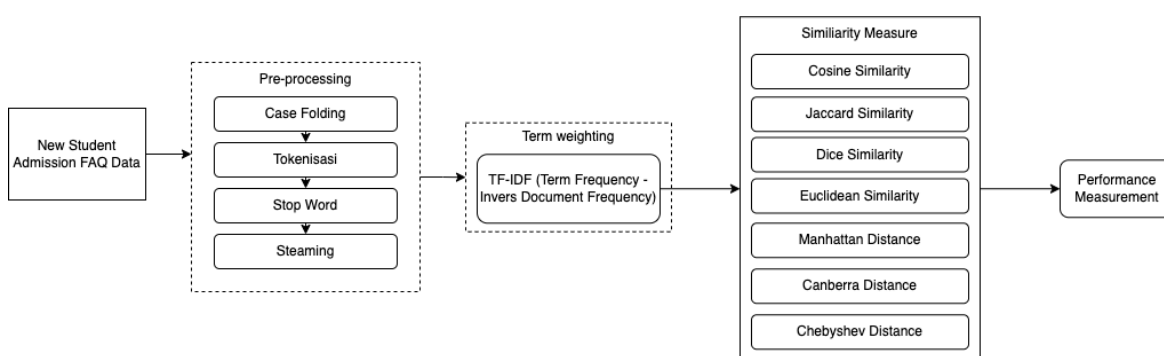


Figure 1 Research Stages

2.1 Pre-processing

Data are cleaned and normalized during pre-processing so that they can be used. Case folding, tokenizing, stopping the word, and steaming are the stages of preprocessing [10]. The process of changing all letters to lowercase is known as case folding. The objective of this is to standardize all letters. "Perkenalan Nama Saya ARIF Mudi" to "perkenalan nama saya arif mudi " is one example of a folding case. The process of breaking up large sentences into smaller ones (words), or tokens, is referred to as tokenizing in Lexical Analysis. "perkenalan nama saya arif mudi" is broken down into "perkenalkan," "nama," "saya," "arif," and "mudi" as an example of tokenization. Stopword is a process for filtering out tokens that aren't needed. ["perkenalkan," "nama," "saya," "arif," "mudi"] to ["perkenalkan," "nama," "arif," "mudi"] are two examples of stopwords processes. Tokens are returned to their root word through steaming. For instance, "kenal" replaces "perkenalkan".

2.2 Term Weighting

The weight of each existing token is determined through weighting. This study employs TF-IDF, or term frequency-inverse document frequency, for the weighting. A statistical measure of how relevant words are to documents in an existing collection is known as TF-IDF. The term frequency and the inverse document frequency create the TF-IDF. The number of token events in the document set is referred to as term frequency (TF), and TF places a significant emphasis on general (often out) tokens. Equation 1 is used to calculate TF [11]. Tokens with a high value that occur infrequently are counted using the inverse document frequency (IDF) method. The IDF uses tokens that only appear occasionally. Equation 2 is used to calculate IDF [11]. To provide an optimal weight value between tokens that appear frequently and tokens that appear infrequently, TF-IDF are combined. Equation 3 [12] is utilized when calculating TF-IDF.

$$TF = \frac{f_{tfd}}{t_{ftd}} \quad (1)$$

$$IDF = \log\left(\frac{ndf}{td}\right) \quad (2)$$

$$W = TF * IDF \quad (3)$$

where TF is the weight of the frequency of the resulting term. The number of times a term appears in the document is called f_{tfd} . The total number of terms that can be found in any given document is called t_{ftd} . The weight of the final inverse document frequency is called IDF. The number of documents with term occurrences is called ndf . The sum of all documents is td . The resulting TF-IDF's weight is W .

2.3 Similarity Measures

The idea of the degree of similarity that exists between two variables or objects is known as the similarity measure. A numerical value (number) that indicates the degree of strength of the association between objects or variables is used as the measure of the level of similarity. The range from 0 to 1 is used. The absence of any similarities between the two items is indicated by the number 0. Number 1 demonstrates the very high degree of similarity between the two things [13]. Cosine similarity, jaccard distance, dice similarity, euclidean distance, Manhattan distance, canberra distance, and Chebyshev distance are the similarity measures used in this study.

2.3.1 Cosine Similarity

Based on the cosine angle, the cosine similarity measure measures how similar two vectors are and determines whether their directions are the same. Cosine similarity can be determined using equation 4 [14].

$$sim(x, y) = \frac{x \cdot y}{||x|| ||y||} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (4)$$

$x = (x_1, x_2, x_3, \dots, x_n)$ and $y = (y_1, y_2, y_3, \dots, y_n)$, where x and y are vector numbers from text or a sentence. n is the length of the vectors x and y . The sentence number index is called i , $i = (1, 2, 3, \dots, n)$. Document vectors 1 or x are represented by the numbers (4, 0, 2, 1, 7, 0, 0, 1, 0, 2). Document vectors 2 or y are represented by the numbers (3, 1, 0, 0, 5, 1, 1, 0, 0, 3, 1, 3). The following calculation reveals that the cosine similarity (x, y) is 0.85.

$$x \cdot y = 4 \times 3 + 0 \times 1 + 2 \times 0 + 1 \times 0 + 7 \times 5 + 0 \times 1 + 0 \times 1 + 1 \times 0 + 1 \times 0 + 1 \times 3 + 0 \times 1 + 2 \times 3 = 56$$

$$||x|| = \sqrt{4^2 + \dots + 2^2} = 8,8$$

$$||y|| = \sqrt{3^2 + \dots + 3^2} = 7,5$$

$$sim(x, y) = \frac{56}{8,8 \times 7,5} = \frac{56}{66} = 0,85$$

2.3.2 Jaccard similarity

Jaccard similarity uses the similarity between sets to determine the degree of similarity between two vectors. This indicates the division of the two data sets' unions [15]. The equation is calculated using Jaccard similarity by comparing two vectors where each document will count the same number of words. A document with a higher similarity value will also have a higher similarity value [16]. A method for calculating jaccard similarity is Equation 5 [17].

$$Jsim(x, y) = \frac{x \cap y}{x \cup y} \quad (5)$$

where y is equal to $(y_1, y_2, y_3, \dots, y_n)$ and x is equal to $(x_1, x_2, \dots, x_n)$. The intersection of the data sets is denoted by the symbol $\cap$, and the symbol $\cup$ denotes a combination of two data.

2.3.3 Dice Similarity

Dice Similarity determines the degree of similarity between the vector data by dividing the number of vector values by the number of intersection values between vector 1 and the other vectors. Equation 6 [18] can be used to calculate dice similarity. In this equation, x is equal to $(x_1, x_2, x_3, \dots, x_n)$, y is equal to $(y_1, y_2, y_3, \dots, y_n)$, and $\cap$ is the intersection of the vector data.

$$Dsim(x, y) = \frac{2|x \cap y|}{|x| + |y|} \quad (6)$$

2.3.4 Euclidean Similarity

The Euclidean distance, as determined by the Pythagorean theorem, is the distance between points on straight lines. Equation 8 is used in calculations to determine the similarity of the vectors because Euclidean distance does not identify similarities between two vectors. This is because Equation 7 is a method for determining Euclidean distance.

$$Ed(x \rightarrow y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (7)$$

$$Esim(x, y) = \frac{1}{1 + Ed(x \rightarrow y)} \quad (8)$$

x is a data vector $(x_1, x_2, x_3, \dots, x_n)$, y is a data vector $(y_1, y_2, y_3, \dots, y_n)$, and i is an index on a vector $(1, 2, 3, \dots, n)$. $Ed(x \rightarrow y)$ is the Euclidean distance between x and y . $Esim(x, y)$ is the Euclidean similarity vector between x and y .

2.3.5 Manhattan Distance

The Manhattan distance, whose calculations are carried out by calculating the sum of all differences between the distances of the two objects, is used to measure distance [19]. The distance value is absolute. The Manhattan distance can be determined using equation 9 [20]. Equation 10 addresses the change from distance to closeness.

$$Md(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (9)$$

$$Msim(x, y) = \frac{1}{1 + Md(x, y)} \quad (10)$$

Where $Md(x, y)$ is the Manhattan distance between x and y and $Msim(x, y)$ is the Manhattan similarity between x and y . x is a data vector with the values $(x_1, x_2, x_3, \dots, x_n)$ and y is a data vector with the values $(y_1, y_2, y_3, \dots, y_n)$. The index of the data vector and its length (n) are equal to $i = (1, 2, 3, \dots, n)$.

2.3.6 Canberra Distance

Godfrey N. introduced the Canberra distance in 1966 and refined it in 1967 as a numerical measure of the distance between pairs of points in a vector space [21]. The distance from Canberra can be calculated using equation 11 [20]. A change in distance to similarity from 1 to 0 is represented by equation 12.

$$CanD(x, y) = \sum_{i=1}^n \frac{|x_i - y_i|}{|x_i| + |y_i|} \quad (11)$$

$$CanSim(x, y) = \frac{1}{1 + CanD(x, y)} \quad (12)$$

$CanD(x, y)$ is the Canberra distance between x and y , and $CanSim(x, y)$ is the Canberra similarity between x and y . x is a data vector with the values $(x_1, x_2, x_3, \dots, x_n)$ and y is a data vector with the values $(y_1, y_2, y_3, \dots, y_n)$. The index of the data vector and its length (n) are equal to $i = (1, 2, 3, \dots, n)$.

Table 1 Confusion Matrix

	Prediction		
		Positive	Negative
	Actual	Positive	Negative
		True Positive (TP)	False Positive (FP)
		False Negative (FN)	True Negative (TN)

Table 2 Information Data and Labels

Information	Labels
Panduan pendaftaran mahasiswa baru dapat dilihat di link berikut ini (The new student registration guide can be seen at the following link) https://pmb.universitaspahlawan.ac.id/site/contentmenu?id=8	Panduan (Guide)
Akreditasi program studi S1 Pendidikan Guru Pendidikan Anak Usia Dini (PG PAUD) yaitu B (The S1 Early Childhood Education Teacher Education (PG PAUD) study program, specifically B, has been granted accreditation)	paud
Jalur masuk Skor TOEFL merupakan jalur bagi siswa yang mempunyai sertifikat tes TOEFL dengan skor 450 - 550. * Skor TOEFL >550 Potongan Uang Pembangunan 100%. * Skor TOEFL 526-549 Potongan Uang Pembangunan 75%. * Skor TOEFL 500-525 Potongan Uang Pembangunan 50%. * Skor TOEFL 450-499 Potongan Uang Pembangunan 25%. (TOEFL score entry path is a pathway for students who have a TOEFL test certificate with a score of 450 - 550. * TOEFL score > 550, 100% Development Fee scholarship. * TOEFL score 526-549, 75% Development Fee scholarship. * TOEFL score 500-525, 50% Development Fee scholarship. * TOEFL score 450-499, 25% Development Fee scholarship.)	Jalurtoefl (toefl)
akreditasi program studi S1 Kewirausahaan yaitu Baik (The S1 Entrepreneurship program has "good" accreditation.)	kewirausahaan (entrepreneurship)
Tes Masuk UP gelombang 1: 27 Juni 2022 (Phase 1 of the UP Entrance Exam:27 June 2022)	tesmasuk (entrancetest)

2.3.7 Chebyshev distance

A technique for determining the absolute maximum value between two objects is the Chebyshev distance. The Chebyshev distance can be calculated using equation 13 [20]. A change from distance to similarity in the range 0 to 1 is represented by equation 14.

$$ChebD(x, y) = \max_i(|x_i - y_i|) \quad (13)$$

$$ChebSim(x, y) = \frac{1}{1 + ChebD(x, y)} \quad (14)$$

2.4 Performance Measurement

A confusion matrix is used in this study for performance measurement. The concept of the confusion matrix is depicted in Table 1 [22].

The achievement of obtaining the desired response is true positive. False Positive gets a response, but not in the way that was expected. The achievement of receiving an incorrect response to a question input is known as false negative. The failure to provide information based on incorrect question input data is a true negative. The F1 Score, precision, recall, and accuracy are the measurements used. According to research [1] [23] [24], the formulas used are equations 15, 16, and 18.

$$F1 - Score = \frac{2 \times (Precision \times recall)}{Precision + recall} \quad (15)$$

$$Precision = \frac{TP}{TP + FP} \quad (16)$$

$$Recall = \frac{TP}{TP + FN} \quad (17)$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (18)$$

Table 3 New Student Questions

Question	Label
Adakah panduan untuk melakukan pendaftaran mahasiswa baru di UP (Is there a guide on how to register new UP students?)	panduan (Guide)
Apa akreditasi S1 PG PAUD (What accreditation does S1 PG PAUD hold?)	paud
Jalur masuk dengan skor TOEFL itu apa ya (With a TOEFL score, what is the path to entry?)	jalurtoefl (toefl)
Apa akreditasi S1 Kewirausahaan (What is S1 Entrepreneurship's accreditation?)	kewirausahaan (entrepreneurship)
Kapan tes masuk untuk gelombang pertama ya (When is the first stage entrance exam?)	tesmasuk (entrancetest)

Table 4 Pre-processing of Data

Pertanyaan	Hasil <i>Pre-processing</i>
adakah panduan untuk melakukan pendaftaran mahasiswa baru di UP (is there an aide for enlisting new understudies at UP)	pandu laku daftar mahasiswa up (guide do student register up)
apa akreditasi S1 PG PAUD (What accreditation does S1 PG PAUD hold?)	akreditasi s1 pg paud (accreditation S1 PG PAUD)
jalur masuk dengan skor TOEFL itu apa ya (What is the entrance path with a TOEFL score?)	jalur masuk skor toefl (toefl score entry path)
apa akreditasi S1 Kewirausahaan (What is S1 Entrepreneurship's accreditation?)	akreditasi s1 kewirausahaan (accreditation S1 Entrepreneurship)
kapan tes masuk untuk gelombang pertama ya (When exactly is the first stage entrance examination?)	tes masuk gelombang (stage entrance examination)

3. RESULTS AND DISCUSSION (10 PT)

This study makes use of information about new students and the accreditation of study programs at Universitas Pahlawan Tuanku Tambusai. There are 41 pieces of information used. Information and labels for that information make up data. A sample of the data used in this study can be found in Table 2.

According to Table 2, prospective students frequently have inquiries regarding accreditation, entry routes, registration guidelines, and universitas pahlawan entrance tests. A sample of questions from prospective new students can be found in Table 3. The questions and information data are first preprocessed. The goal of preprocessing is to get rid of terms that are used a lot or have less meaning. The outcome of the pre-processing phase is shown in Table 4.

Table 4 displays the questions and information from the preprocessing results. When determining the degree of similarity, the results of the preprocessing are used as input. This paper compares cosine, jaccard, dice, euclidean, Manhattan, canberra, and chebyshev similarity measures. The aftereffects of closeness are positioned. The outcomes with the highest ranking and the highest threshold will be used. The similarity ranges from one to zero. A value of one indicates very similarity or identity. Nothing about the value 0 is the same. This test has 0.1, 0.2, 0.3, 0.4, and 0.5 thresholds. The results of the tests are shown in Table 5.

Table 5 displays the test scenarios for the cosine, jaccard, dice, euclidean, Manhattan, Canberra, and Chebyshev similarity chatbots. The cosine similarity chatbot's results demonstrate that the cosine similarity threshold of 0.3 has produced consistent results. This demonstrates that the f1-score of the cosine similarity method improves with decreasing threshold. At the thresholds of 0.1, 0.2, and 0.3, the best recall is 100 percent, but the success precision is lower than at the thresholds of 0.4 and 0.5. This demonstrates that not all of the obtained data are accurate. The data that have been obtained are correct because the threshold condition of 0.4 indicates that precision and recall are balanced.

The jaccard similarity chatbot's results demonstrate that the f1-score's success has been consistent at the 0.2 threshold. The chatbot was able to get an answer with a recall rate of over 90% and a precision of over 75% at the thresholds of 0.1 and 0.2. The best precision is 81.08 percent, and the F1-score at the threshold is the same, which is 86.96 percent. This demonstrates that eighty percent of the chatbot's ninety percent success rate in obtaining an answer matches the anticipated response.

Table 5 Comparison of Similarity Methods' Outcomes

Metode	Threshold	F1-Score	Presisi (%)	Recall (%)	Akurasi (%)
Cosine	0.5	80	96,3	68,42	68,29
	0.4	85,29	85,29	85,29	75,61
	0.3	86,96	76,92	100	78,05
	0.2	86,96	76,92	100	78,05
	0.1	86,96	76,92	100	78,05
Jaccard	0.5	65,52	100	48,72	51,22
	0.4	76,19	100	61,54	63,41
	0.3	85,58	93,33	75,68	73,17
	0.2	86,96	81,08	93,75	78,05
	0.1	86,96	76,92	100	78,05
Dice	0.5	83,58	96,55	73,68	73,17
	0.4	86,96	85,71	88,24	78,05
	0.3	86,96	76,92	100	78,05
	0.2	86,96	76,92	100	78,05
	0.1	86,96	76,92	100	78,05
Euclidean	0.5	50	100	33,33	36,59
	0.4	72,13	100	56,41	58,54
	0.3	85,71	76,92	96,77	75,61
	0.2	85,71	76,92	96,77	75,61
	0.1	85,71	76,92	96,77	75,61
Manhattan	0.5	9,76	100	5,13	9,76
	0.4	40,82	100	25,64	29,27
	0.3	47,06	100	30,77	34,15
	0.2	47,06	100	30,77	34,15
	0.1	62,07	90	47,37	46,34
Canberra	0.5	0	0	0	4,88
	0.4	14,29	100	7,69	12,20
	0.3	30,43	100	17,95	21,95
	0.2	30,43	100	17,95	21,95
	0.1	47,06	100	30,77	34,15
Chebyshev	0.5	66,67	50,00	100	51,22
	0.4	66,67	50,00	100	51,22
	0.3	66,67	50,00	100	51,22
	0.2	66,67	50,00	100	51,22
	0.1	66,67	50,00	100	51,22

At a threshold of 0.4, the dice similarity chatbot demonstrates that the method produces consistent f1-score results. The dice similarity chatbot was successful in obtaining all of the responses at the 0.1, 0.2, and 0.3 thresholds, and it was also successful in providing the desired response, which was 76.92 percent. The results of the threshold 0.4 chatbot dice similarity are very satisfying. The f1-score, recall, and precision above 85%, or, more precisely, 86.96%, 88.24%, and 85.71 percent, respectively, demonstrate this.

The chatbot attains a consistent f1-score at the 0.3 threshold, as shown by the euclidean similarity of the chatbot. At the thresholds of 0.3, 0.2, and 0.1, the f1-score, recall, and precision values are, respectively, 85.71 percent, 96.77 percent, and 76.92 percent. Between 0.4 and 0.3, the Euclidean distance chatbot has a long recall distance. This demonstrates that the typical success value for getting answers is between 0.3 and 0.4. Only half of the questions asked by threshold 0.4 were answered.

The Chatbot Manhattan similarity demonstrates that chatbots produce unfavorable outcomes. The f1-score of no more than 70% demonstrates this. The f1-score for threshold 0.1 was only 62.07 percent. Less than half of the questions asked by chatbots have a threshold of 0.1. The recall at the 0.2 and 0.3 thresholds, which is 30.77 percent, demonstrates that the similarity score calculated with Manhattan is 30% between 0.3 and 0.4. The chatbot's similarity to Canberra demonstrates that it is still not up to par. The Canberra Similarity chatbot was successful in answering 30% of the questions. The f1-score value of 47.06 percent is the highest. The similarity score that Canberra got, which was 31%, is between 0.3 and 0.4. A recall of 0% at a threshold of 0.5 indicates that the highest threshold that can be used is 0.4.

The f1-score, precision, and recall success rates were 66.67 percent, 50 percent, and 100 percent, respectively, as demonstrated by the Chebyshev chatbot. This demonstrates that the chatbot was able to

obtain responses in excess of one hundred percent, despite having only fifty percent precision. Because the chatbot's similarity score is higher than 0.5, remember everything. The Chebyshev chatbot does not perform at its best, with a f1-score of 66.67 percent and a precision of 50%.

On chatbots with dice similarity, the overall test results yield optimal results. The obtained optimal f1-score, which is above 80% for all thresholds, indicates this optimal result. Cosine, jaccard, and dice similarity are examples of optimal similarity. Chebyshev, Manhattan, and Canberra similarity are examples of not-optimal similarity. At the thresholds of 0.1, 0.2, and 0.3, recall on dice and cosine yields optimal results of 100%. Good precision, exceeding 75%, was found in conjunction with the recall. This is because the chatbot covers new student admissions and accreditation for study programs. Therefore, on the basis of this study, dice similarity or cosine can be utilized in the creation of a chatbot for Universitas Pahlawan Tuanku Tambusai.

4. CONCLUSION (10 PT)

A college's students are an essential component. The college always opens registration for new students each year. Over a thousand prospective new students enroll annually. Because of this, the new student admissions committee is constantly overwhelmed when responding to campus-related questions. As a result, developing a chatbot to assist new students is necessary. The improvement of a chatbot utilizing a recovery model methodology requires the best similitude technique. The New Student Admission Chatbot and the Similarity Method are compared in this study using the Retrieval-Based Concept. The cosine, jaccard, dice, euclidean, Manhattan, canberra, and chebyshev similarity methods are compared. The best test results come from using dice and cosine similarity. On all tested thresholds, dice and cosine similarity achieved an f1-score above 80%. Recall produces extremely optimal results—100 percent. Over 75% of the time, good results are reliably achieved. This demonstrates that the retrieval-model concept can be applied to the creation of chatbots using the dice or cosine similarity method.

REFERENCES

- [1] M. R. A. Prasetya and A. M. Priyatno, "Dice Similarity and TF-IDF for New Student Admissions Chatbot," *RIGGS J. Artif. Intell. Digit. Bus.*, vol. 1, no. 1, pp. 13–18, Jul. 2022, doi: 10.31004/riggs.v1i1.5.
- [2] A. Elcholiqi and A. Musdholifah, "Chatbot in Bahasa Indonesia using NLP to Provide Banking Information," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 14, no. 1, p. 91, Jan. 2020, doi: 10.22146/ijccs.41289.
- [3] G. A. Santos, G. G. de Andrade, G. R. S. Silva, F. C. M. Duarte, J. P. J. Da Costa, and R. T. de Sousa, "A Conversation-Driven Approach for Chatbot Management," *IEEE Access*, vol. 10, pp. 8474–8486, 2022, doi: 10.1109/ACCESS.2022.3143323.
- [4] M. Adam, M. Wessel, and A. Benlian, "AI-based chatbots in customer service and their effects on user compliance," *Electron. Mark.*, vol. 31, no. 2, pp. 427–445, Jun. 2021, doi: 10.1007/s12525-020-00414-7.
- [5] M. D. Illescas-Manzano, N. V. López, N. A. González, and C. C. Rodríguez, "Implementation of chatbot in online commerce, and open innovation," *J. Open Innov. Technol. Mark. Complex.*, vol. 7, no. 2, p. 125, May 2021, doi: 10.3390/joitmc7020125.
- [6] S. Han and M. K. Lee, "FAQ chatbot and inclusive learning in massive open online courses," *Comput. Educ.*, vol. 179, p. 104395, Apr. 2022, doi: 10.1016/j.compedu.2021.104395.
- [7] P. Amiri and E. Karahanna, "Chatbot use cases in the Covid-19 public health response," *J. Am. Med. Informatics Assoc.*, vol. 29, no. 5, pp. 1000–1010, 2022, doi: 10.1093/jamia/ocac014.
- [8] R. Agarwal and M. Wadhwa, "Review of State-of-the-Art Design Techniques for Chatbots," *SN Comput. Sci.*, vol. 1, no. 5, p. 246, Sep. 2020, doi: 10.1007/s42979-020-00255-3.
- [9] A. Soufyane, B. A. Abdelhakim, and M. Ben Ahmed, "An intelligent chatbot using NLP and TF-IDF algorithm for text understanding applied to the medical field," in *Advances in Science, Technology and Innovation*, Springer, 2021, pp. 3–10. doi: 10.1007/978-3-030-53440-0_1.
- [10] N. P. Ririanti and A. Purwinarko, "Implementation of Support Vector Machine Algorithm with Correlation-Based Feature Selection and Term Frequency Inverse Document Frequency for Sentiment Analysis Review Hotel," *Sci. J. Informatics*, vol. 8, no. 2, pp. 297–303, 2021, doi: 10.15294/sji.v8i2.29992.
- [11] N. S. Mohd Nafis and S. Awang, "An Enhanced Hybrid Feature Selection Technique Using Term Frequency-Inverse Document Frequency and Support Vector Machine-Recursive Feature Elimination for Sentiment Classification," *IEEE Access*, vol. 9, no. M1, pp. 52177–52192, 2021, doi: 10.1109/ACCESS.2021.3069001.
- [12] M. Z. Naeem, F. Rustam, A. Mehmood, Mui-zzud-din, I. Ashraf, and G. S. Choi, "Classification of movie reviews using term frequency-inverse document frequency and optimized machine learning

- algorithms,” *PeerJ Comput. Sci.*, vol. 8, p. e914, Mar. 2022, doi: 10.7717/PEERJ-CS.914.
- [13] D. Ifenthaler, “Measures of Similarity,” in *Encyclopedia of the Sciences of Learning*, Boston, MA: Springer US, 2012, pp. 2147–2150. doi: 10.1007/978-1-4419-1428-6_503.
- [14] R. Singh and S. Singh, “Text Similarity Measures in News Articles by Vector Space Model Using NLP,” *J. Inst. Eng. Ser. B*, vol. 102, no. 2, pp. 329–338, 2021, doi: 10.1007/s40031-020-00501-5.
- [15] H. Khojamli and J. Razmara, “Survey of similarity functions on neighborhood-based collaborative filtering,” *Expert Syst. Appl.*, vol. 185, no. March, p. 115482, 2021, doi: 10.1016/j.eswa.2021.115482.
- [16] S. Bag, S. K. Kumar, and M. K. Tiwari, “An efficient recommendation generation using relevant Jaccard similarity,” *Inf. Sci. (Nij.)*, vol. 483, pp. 53–64, May 2019, doi: 10.1016/j.ins.2019.01.023.
- [17] S. Varma, S. Shivam, A. Thumu, A. Bhushanam, and D. Sarkar, “Jaccard Based Similarity Index in Graphs: A Multi-Hop Approach,” in *2022 IEEE Delhi Section Conference (DELCON)*, Feb. 2022, pp. 1–4. doi: 10.1109/DELCON54057.2022.9753316.
- [18] A. Bahrul Khoir, H. Qodim, B. Busro, and A. Rialdy Atmadja, “Implementation of rabin-karp algorithm to determine the similarity of synoptic gospels,” *J. Phys. Conf. Ser.*, vol. 1175, no. 1, p. 012120, Mar. 2019, doi: 10.1088/1742-6596/1175/1/012120.
- [19] R. Suwanda, Z. Syahputra, and E. M. Zamzami, “Analysis of Euclidean Distance and Manhattan Distance in the K-Means Algorithm for Variations Number of Centroid K,” *J. Phys. Conf. Ser.*, vol. 1566, no. 1, p. 012058, Jun. 2020, doi: 10.1088/1742-6596/1566/1/012058.
- [20] S. Liaquat, M. F. Zia, O. Saleem, Z. Asif, and M. Benbouzid, “Performance analysis of distance metrics on the exploitation properties and convergence behaviour of the conventional firefly algorithm,” *Appl. Soft Comput.*, vol. 126, p. 109255, Sep. 2022, doi: 10.1016/j.asoc.2022.109255.
- [21] M. Faisal, E. M. Zamzami, and Sutarman, “Comparative Analysis of Inter-Centroid K-Means Performance using Euclidean Distance, Canberra Distance and Manhattan Distance,” *J. Phys. Conf. Ser.*, vol. 1566, no. 1, p. 012112, Jun. 2020, doi: 10.1088/1742-6596/1566/1/012112.
- [22] A. M. Priyatno, “Spammer Detection Based on Account, Tweet, and Community Activity on Twitter,” *J. Ilmu Komput. dan Inf.*, vol. 13, no. 2, pp. 97–107, Jul. 2020, doi: 10.21609/jiki.v13i2.871.
- [23] A. M. Priyatno, M. M. Muttaqi, F. Syuhada, and A. Z. Arifin, “Deteksi bot spammer twitter berbasis time interval entropy dan global vectors for word representations tweet’s hashtag,” *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 5, no. 1, pp. 37–46, Jan. 2019, doi: 10.26594/register.v5i1.1382.
- [24] A. M. Priyatno, F. M. Putra, P. Cholidhazia, and L. Ningsih, “Combination of extraction features based on texture and colour feature for beef and pork classification,” *J. Phys. Conf. Ser.*, vol. 1563, no. 1, p. 012007, Jun. 2020, doi: 10.1088/1742-6596/1563/1/012007.