Chatbot for Information Service of New Student Admission Using Multinomial Naïve Bayes Classification and TF-IDF Weighting

Khoirida Aelani[1,*] Gugi Gustaman[1]

[1]*Information Systems, STMIK "AMIKBANDUNG", Bandung, Indonesia*
[*]*Corresponding author. Email: khoirida@stmik-amikbandung.ac.id*

ABSTRACT

New student admission is a process where prospective students need the information to decide which higher education institution they will enroll. Live chat on the institution website is one of the reliable information sources to find information about the institution. Most higher education institutions have their website but not every single of them has implemented a live chat feature on the website. Live chat requires humans to answer website visitors' questions. However, there are limitations in humans to always be able to respond and provide answers accurately. A chatbot is a machine learning implementation that can be applied to overcome these limitations. Natural Language Processing (NLP) concept can help chatbots translate human language and help the computer understand what humans mean in their language. The machine learning model that is used for classification is Multinomial Naïve Bayes with the help of term weighting using TF-IDF. With the classification model, prospective students' questions can be classified based on their intents, but the model needs labeled questions as the training data. Chatbot with quality training data set and 24/7 service availability makes prospective students' questions can be answered quickly anytime and anywhere. In this research, 1.330 questions were gathered as training data and grouped into 17 intents. More than 90% of questions predicted correctly using K-Fold Cross Validation, but only 65% when the chatbot is tested by website visitors due to less clean and less complete training data set that obviously can be improved in the future.

Keywords: *Chatbot, Multinomial Naïve Bayes, Term Frequency - Inverse Document Frequency (TF-IDF), Machine Learning.*

1. INTRODUCTION

New Student Admission (NSA) is an activity that cannot be separated from a higher education institution's business process. NSA is regularly held by the institutions to add new students. NSA must be planned and utilized well for getting the desired result for the institutions. Not only for the institutions, but NSA is also very important to prospective students before they enroll.

Every higher education institution has a website that represents the institution. Based on research in 2019, a website is the most influential resource to get information about the institute [1]. However, a website is a one-way medium that is designed with different user interfaces from one another and that is why users must learn every single website they just visited. Research shows that users' familiarity makes users feel comfortable in navigating the website, so they are sure that the website has reliable information [2]. It is necessary to add another medium to answer the needs of prospective students better, quicker, and more relevant to the information they need.

One of the media that can be used to answer their needs is through chat box. Nowadays, people use chat box regularly. The conversational interface similarity of one and another chat platforms makes even technologically backward people use them easily. Moreover, there is a live chat that is generally embedded on websites for visitors to ask for information about products or services the website owners provide. Prospective students can use live chat to get more necessary information about the higher education institutions they want to enroll. But human is required to answer prospective students' questions and it requires additional costs. Furthermore, research shows that response time and information quality are some of the most important factors in a satisfying live chat experience [3]. As we know, humans make mistakes and need rest. It would be problems of information quality and availability for the live chat service. However, those problems can be solved by a chatbot when computer answers every single question quickly and accurately whenever needed. In previous research, 68% of people use chatbots because of the productivity they can produce. A chatbot can provide ease, speed and convenience also obtain help and information for users [4]. Moreover, the time needed to interact with a chatbot is much faster than the time needed to interact with the website [2].

In this research, the language that is being used is Bahasa Indonesia. The system only implements Natural Language Understanding (NLU) and classifies the questions. The random answer will be selected from the database based on the classified intents. The chatbot will only predict questions as a single document.

2. LITERATURE REVIEW

2.1. Term Frequency – Inverse Document Frequency

Term Frequency – Inverse Document Frequency (TF-IDF) is a term weighting method that calculates information amount by its occurrence probability [5]. The calculation is shown in below Equation (1).

$$TFIDF_{t,d} = TF_{t,d} \times IDF_{t,D} \tag{1}$$

Where:

TF$TF_{t,d}$: normalized term frequency of term *t* in document *d*
$IDF_{t,d}$: inverse document frequency of term *t* in all documents

2.1.1. Term Frequency

Term Frequency is the number of term *t* that exists in a document *d* [6]. In a normalized form, it is notated as Equation (2) shows below where the most used *K* value is 0.5.

$$TF_{t,d} = K + (1 - K) \times \left(\frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}} \right) \tag{2}$$

Where:

$TF_{t,d}$: normalized term frequency of term t in document d
K : 0,5 is used, so the term frequency of term *t* will be between 0,5 to 1
$f_{t,d}$: number of term *t* in document *d*
max{$f_{t',d}$ *: t' '∈d}* : number of term *t'*, where *t'* is the most word in document *d*

2.1.2. Inverse Document Frequency

Inverse Document Frequency (IDF) indicates how common or rare a term is in all documents by basically counting the number of documents that have term *t* [7]. It is a common technique used in information retrieval.

$$IDF_t = \log\left(\frac{D}{df_t}\right) \tag{3}$$

Where:

$IDF_{t,d}$: inverse document frequency of term *t* in all documents
D : number of documents in the training set
df_t : number of documents in the training set having term *t*

2.2. Multinomial Naïve Bayes

Multinomial Naïve Bayes is a text classification model which is a supervised and probabilistic learning method [8]. It is an extension of Naïve Bayes algorithm which Naïve Bayes is used only for two-classes classification, but Multinomial Naïve Bayes is used for two or more classes classification. In this research, TF-IDF weight is being used in the Multinomial Naïve Bayes classification as Equation (4) shows [9].

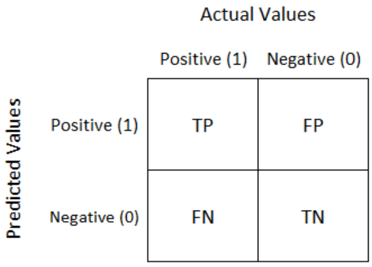
$$P(x|w) = \frac{(\sum TFIDF_{x,w}) + \alpha}{\sum TF_w + \alpha \times V} \tag{4}$$

Where:

P(x|w) : Probability of term x on *intent* w
$\Sigma\, TFIDF_{x,w}$: TF-IDF weight total of term x on *intent* w
$\Sigma\, TF_w$: Total of all terms TF on *intent* w
α : *Laplace Smoothing*
V : Vocabulary size on the training data set

2.3. Confusion Matrix

Confusion Matrix is a matrix used to measure the performance of machine learning classification with two or more classes. As Figure 1 shows, it is represented as rows and columns where the diagonal (from top-left to bottom-right) is the correct prediction and the rest is incorrect predictions [10]. The number of rows or columns is determined by the number of classes in the classification.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 1 Confusion matrix

Where:

TP: Number of correct predictions on positive class
FP: Number of incorrect predictions on positive class
TN: Number of correct predictions on negative class
FN: Number of incorrect predictions on negative class

Some measuring tools are proposed and used to evaluate classification model performance. Below are some popular measuring tools for classification.

1. Accuracy describes how accurate the model in predicting correctly as shown in Equation (5).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

2. Precision, the alternative of accuracy, is the total of correct predictions divided by the total of positive predictions either they are true or false in a class as shown in Equation (6).

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

3. Recall describes the success of the model in retrieving information as shown in Equation (7).

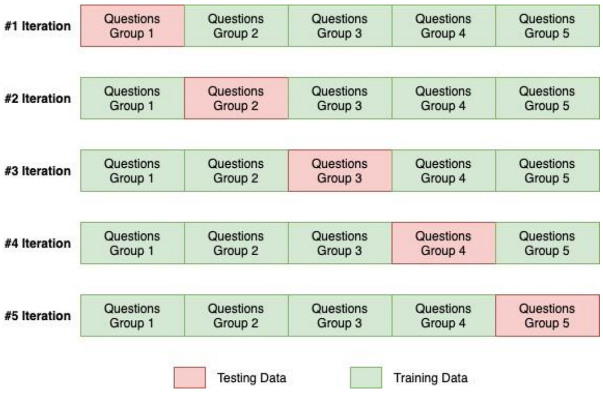
$$Recall = \frac{TP}{TP + FN} \tag{7}$$

4. F-1 Score, the harmonic average of precision and recall in a class as shown in Equation (8).

$$Score\ F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

2.4. K-Fold Cross-Validation

Cross-validation is a method that is used to test a machine learning model. It separates training data set into training data and testing data [11]. There are two techniques to do cross-validations, they are Train_Test Split and K-Fold. K-Fold Cross Validation separates the training data set into groups and iterates through the groups until all groups are tested to the rest groups (training data) as Figure 2 shows.



#1 Iteration	🟥 Questions Group 1	🟩 Questions Group 2	🟩 Questions Group 3	🟩 Questions Group 4	🟩 Questions Group 5
#2 Iteration	🟩 Questions Group 1	🟥 Questions Group 2	🟩 Questions Group 3	🟩 Questions Group 4	🟩 Questions Group 5
#3 Iteration	🟩 Questions Group 1	🟩 Questions Group 2	🟥 Questions Group 3	🟩 Questions Group 4	🟩 Questions Group 5
#4 Iteration	🟩 Questions Group 1	🟩 Questions Group 2	🟩 Questions Group 3	🟥 Questions Group 4	🟩 Questions Group 5
#5 Iteration	🟩 Questions Group 1	🟩 Questions Group 2	🟩 Questions Group 3	🟩 Questions Group 4	🟥 Questions Group 5

🟥 Testing Data 🟩 Training Data

Figure 2 K-fold cross-validation illustration

3. RESEARCH METHODS

3.1. Requirements Gathering

As this research uses waterfall SDLC, so the first thing to do is gathering requirements. As a supervised learning method, Multinomial Naïve Bayes needs training data set to classify incoming questions from prospective students. It can be achieved by asking student admission staff what kind of questions (intents) prospective students usually ask when they want to get information about new student admission. The process result is the questions' intents. Then, unique questions of the intents must be gathered to complete the training data. The number of questions on each intent is better to be balanced to produce better classification results.

3.2. Analysis

There are two main processes the chatbot does in this research. Those are the training process and the query process.

3.2.1. Training

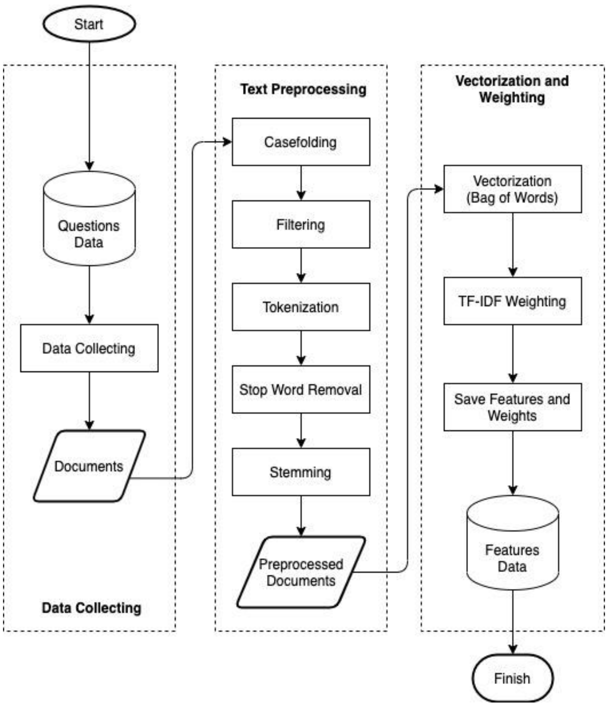


Figure 3 Training process flow

As Figure 3 shows, first, training data is gathered and prepared to be trained to the chatbot. The result of this phase is a collection of questions with every intent. The collection is then passed to the text preprocessing phase. In the text preprocessing phase, each question from the collection is preprocessed to extract features from them. In this research, the subprocesses on this phase are ordered below.

3.2.1.1. Case Folding

Case folding means converting the questions into lower case. For example, *Berapa* (how much) will be converted to *berapa* to vectorize the terms better.

3.2.1.2. Filtering

In this subprocess, punctuations and non-alphanumeric characters such as period, question mark, and other characters will be removed from case-lowered questions as they mean nothing in the classification model of this research.

3.2.1.3. Tokenization

Tokenization is the most important subprocesses of text preprocessing where the question is chunked into terms that are originally separated with space.

3.2.1.4. Stop Word Removal

In Indonesian language, things like pronouns and conjunctions that often appear will also be removed as they will affect the weight calculation, such as *saya* (I/me), *kamu* (you), *dan* (and), *atau* (or), etc.

3.2.1.5. Stemming

The last subprocess of text preprocessing that is used to change the form of terms into their basic form. For example, the term *pendaftaran* (registration) will be changed into “daftar” as its basic form. The purpose is to reduce the number of unique terms that will be processed by the system.

The last phase of the training process is vectorization and term weighting. The terms from text preprocessing are then grouped into different terms and calculated its occurrence.

3.2.2. Query

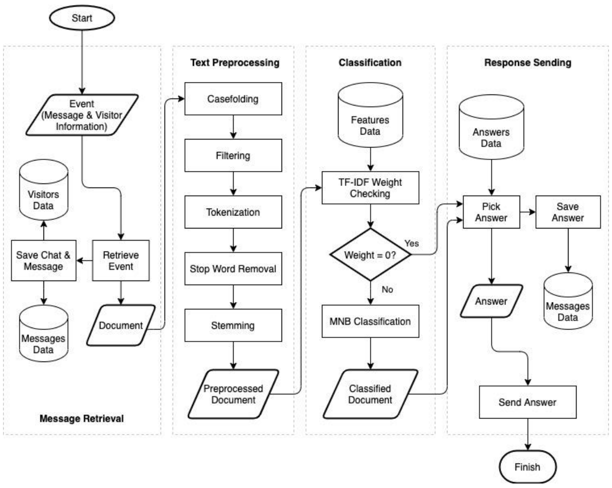


Figure 4 Query process flow

The process to test the training data that is coming from the testing process or website visitor is called a query. As Figure 4 shows, it does the same text preprocessing as the training process phase does. But then each feature from the text preprocessing result of the query will be used to calculate its probability to every registered intent on the training data using Multinomial Naïve Bayes. The classified document will be given a random answer that suits the intent. If the query comes from the testing process, it will be saved in the database as a validation result. If it comes from a website visitor, then the answer will be sent to the visitor through the live chat API.

3.3. Design

The design phase in this research is using Unified Modelling Language (UML). It requires the result of the analysis phase as the input.

3.4. Implementation

In this phase, the chatbot engine is developed using the Object-Oriented Programming (OOP) concept. The Multinomial Naïve Bayes classification and TF-IDF weighting are implemented on the engine creation. After the engine is ready, integration with Live Chat is done through webhooks for the retrieval of website visitor messages and API to send the answers (responses) of the visitor messages.

3.5. Testing

To test the chatbot classification, K-Fold Cross Validation is used with 5 as the K value. The validation result is stored in the database to get the confusion matrix and to get the accuracy, precision, recall, and F-1 score of the classification. Apart from cross-validation that is done internally, a test is also done externally by classifying questions directly from the website visitors.

4. RESULTS

In this research, 1.330 questions are gathered and distributed into 17 intents during the requirements gathering phase as shown in Table 1. Each intent has three answers where one of the three will be randomly given to every incoming question from the website visitors.

Table 1. Intents list

No	Intent	Number of Questions
1	Salaam	55
2	Greetings	68
3	Registration Status	75
4	Registration Procedures	81
5	Registration Fee	68
6	Registration Requirements	76
7	Scholarship Availability	85
8	Campus Location/Address	85
9	Tuition Fee	86
10	List of Majors	81
11	Employee Class Program Availability	69
12	Institution Accreditation	81
13	Majors Accreditation	87
14	Facilities	83
15	Student Organizations	86
16	Student Activity Units	84
17	Ending Conversation	80
Total		1.330

The result of the analysis phase is documented in UML designs. The actors of the system are the website visitors, especially the prospective students and the chatbot itself as shown in Figure 5.

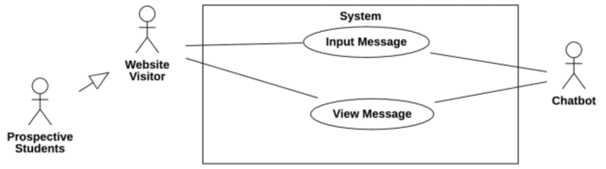


Figure 5 Use case diagram

The entity classes used in the system are shown in Figure 6 as the class diagram.

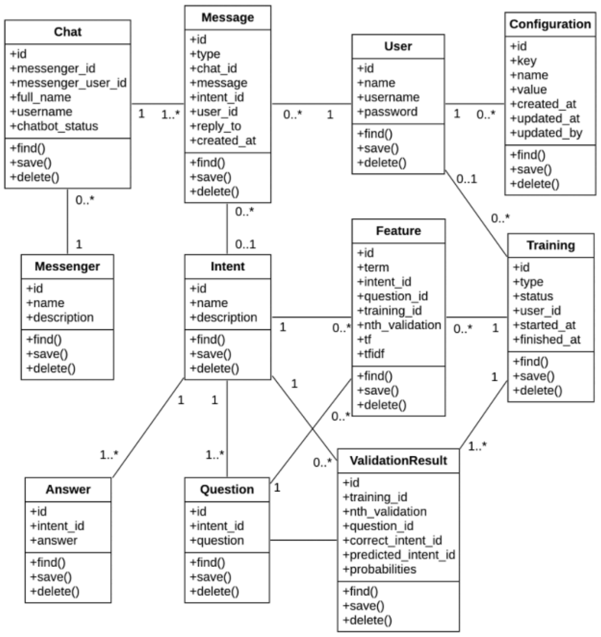


Figure 6 Class diagram of the system entities

After the chatbot is implemented, the training data is trained to chatbot and cross-validated. The result of the training validation result is shown in Table 2.

Table 2. Training cross-validation result

No.	Intent	TP	FN	FP	TN
1	Salaam	52	3	2	1.275
2	Greetings	61	7	7	1.260
3	Registration Status	72	3	7	1.253
4	Registration Procedures	76	5	6	1.248
5	Registration Fee	61	7	5	1.261
6	Registration Requirements	75	1	3	1.251
7	Scholarship Availability	79	6	2	1.244
8	Campus Location/Address	84	1	2	1.244
9	Tuition Fee	78	8	9	1.236
10	List of Majors	70	11	11	1.245
11	Employee Class Program Availability	68	1	0	1.261
12	Institution Accreditation	71	10	3	1.249
13	Majors Accreditation	82	5	15	1.230
14	Facilities	78	5	8	1.241

15	Student Organizations	82	4	2	1.243
16	Student Activity Units	81	3	1	1.245
17	Ending Conversation	76	4	1	1.249
	Total	1.246	84	84	21.235

From the result in Table 2, the accuracy, precision, recall, and F-1 score of the model performance can be calculated as shown in Table 3.

Table 3. Classification model performance

No.	Intent	Accu-racy	Preci-sion	Recall	F-1 score
1	Salaam	0.996	0.963	0.945	0.954
2	Greetings	0.990	0.897	0.897	0.897
3	Registration Status	0.993	0.911	0.960	0.935
4	Registration Procedures	0.992	0.927	0.938	0.933
5	Registration Fee	0.991	0.924	0.897	0.910
6	Registration Requirements	0.997	0.962	0.987	0.974
7	Scholarship Availability	0.994	0.975	0.929	0.952
8	Campus Location/Address	0.998	0.977	0.988	0.982
9	Tuition Fee	0.987	0.897	0.907	0.902
10	List of Majors	0.984	0.864	0.864	0.864
11	Employee Class Program Availability	0.999	1.000	0.986	0.993
12	Institution Accreditation	0.990	0.959	0.877	0.916
13	Majors Accreditation	0.985	0.845	0.943	0.891
14	Facilities	0.990	0.907	0.940	0.923
15	Student Organizations	0.995	0.976	0.953	0.965
16	Student Activity Units	0.997	0.988	0.964	0.976
17	Ending Conversation	0.996	0.987	0.950	0.968
	Macro Average	0.993	0.939	0.937	0.938

Referring to Table 3, it is known that macro averages of the model show that the classification model is very good at predicting questions that come from the training data itself. But the chatbot system will be used publicly, so an external test has been done to check the readiness of the chatbot. In the external test, 170 questions were asked by 31 users (website visitors). The prediction result is shown in Figure 7 as the research predicted 65% (111) questions correctly.

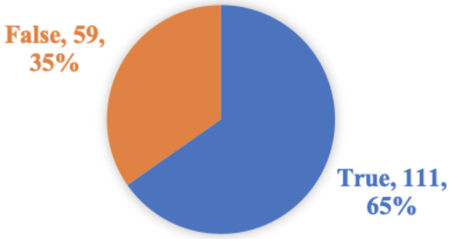


Figure 7 External test prediction result

It is a very different result compared with the internal test using cross-validation. The 59 incorrect predictions are categorized and shown in Figure 8.

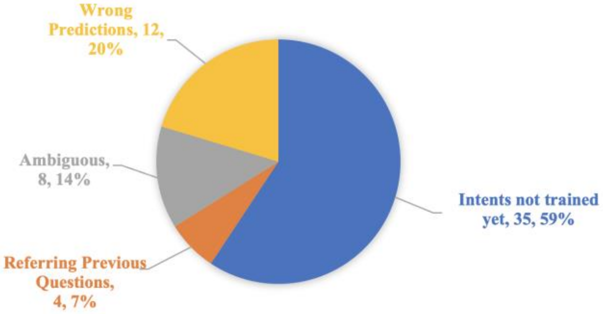


Figure 8 Incorrect predictions

The main cause of the occasions is overfitting. It is a condition where there are one or more terms in each question that have a large weight on predicted intents. Table 4 shows some questions that are being incorrectly predicted and which categories they are in.

Table 4. Some incorrectly predicted questions and their findings

Question	Finding
Apakah semester depan pelaksanaan pembelajaraan sudah pasti offline? (Will there be offline learning next semester?)	Intents not trained yet
Masa studi nya berapa lama? (How long is the study period?)	
Gimana cara daftar beasiswanya? (How to apply the scholarship?)	

Question	Finding
Saya kan bekerja, kira-kira memungkinkan untuk studi di kampus AMIK Bandung? (I am an employee, is it possible for me to study at AMIK Bandung?)	Wrong predictions
Adakah program diploma? (Is there a diploma program?)	Wrong predictions
Apakah ada kelas karyawan? Dan jika ada untuk tahun ada berapa gelombang pendaftarannya dan kapan waktunya? (Are there employee classes? If there are ones for this year, how many registration sessions and when they are opened?)	
Terdapat program apa saja? (What programs are available?)	Ambiguous
Mahasiswa baru (New students)	
Teknik Informatika (Informatics Engineering)	
berlaku sampai tahun berapa? (Until what year is it valid?)	Referring Previous Questions
Dengan jenjang S1 semua? (Are all of them S1/bachelor degree?)	

Referring to Table 4, below are the explanation of the findings.

a. Intents not trained yet

The incorrectly predicted questions are categorized into this finding if their original intents are not trained yet to the chatbot system. The intent of *Gimana cara daftar beasiswanya?* (How to apply for the scholarship?) is intended to ask scholarship registration procedure which is not trained yet to the chatbot.

b. Wrong predictions

This finding shows that the actual intents of the questions are already trained to the system but the chatbot still predicted them to other intents. For example, *saya kan bekerja, kira-kira memungkinkan untuk studi di kampus AMIK Bandung* (I am an employee, is it possible for me to study at AMIK Bandung?) which has intents to ask employee class program availability, but the system predicted it to other or fallback intent.

c. Ambiguous

Incorrectly predicted questions are categorized in this finding if the intent is not clear. It is because the questions consist of a really few words or it is not clear what the users want to ask. The questions like *terdapat program apa saja?* (what programs are available?) and *mahasiswa baru* (new students) don't have really clear intentions.

d. Referring Previous Questions

This finding categorized incorrectly predicted questions that refer to previous questions. But the chatbot currently classifies only individual questions and does not support referring to previous questions yet. A question such as *berlaku sampai tahun berapa?*" (Until what year is it valid?) is referring to the visitor's previous question *peringkat akreditasi teknik informatika* (informatics engineering accreditation rating) and the chatbot only predicted each of them as a single document.

5. CONCLUSIONS AND RECOMMENDATIONS

This research finds that chatbots can overcome human limitations to always be able to respond to prospective students' questions. It quickly classifies questions and gives answers in less than five seconds. By using cross-validation, it predicted more than 90% of questions correctly. Then, the classification works well internally. On live tests by website visitors, this research finds that chatbot only predicted 65% of questions correctly. The rest are predicted incorrectly because the live users test the chatbot with questions with the intents that are not trained yet. They also asked ambiguous questions and questions that refer to their previous questions which are not implemented yet in this research. But then it concludes that a cleaner and more complete training data set produces better classification accuracy.

For future research, another classification method such as logistic regression or any other related method can be implemented to find which method performs better. The chatbot also should be trained with cleaner and more complete data to produces better classification performance. There may be an overfitting case, but it is a general problem in the machine learning classification model. By this way, feature selection can be implemented to reduce the problem occurrences. To provide a better experience, a conversational chatbot can be implemented, so users can ask questions that refer to their previous questions. Furthermore, the chatbot can be integrated into another system such as an academic information system to provide more access to students for getting information about their academic, for example students' GPA.

REFERENCES

[1] Ruffalo Noel Levitz and OmniUpdate, “2019 E-Expectations ® Trend Report,” Cedar Rapids, 2019.

[2] S. Valtolina, B. R. Barricelli, S. Di Gaetano, and P. Diliberto, “Chatbots and Conversational Interfaces: Three Domains of Use,” 2018. Accessed: Jun. 15, 2021. [Online]. Available: http://ceur-ws.org/Vol-2101/paper8.pdf.

[3] G. Mclean and K. Osei-Frimpong, “Examining Satisfaction with The Experience During a Live Chat Service Encounter-Implications for Website Providers.,” 2017. doi: 10.1016/j.chb.2017.08.005.

[4] P. Brandtzaeg and A. Følstad, “Why People Use Chatbots,” in *The 4th International Conference on Internet Science*, 2017, pp. 1–19, doi: 10.1007/978-3-319-70284-1_30.

[5] A. Aizawa, “An information-theoretic perspective of tf–idf measures,” *Inf. Process. Manag.*, vol. 39, no. 1, pp. 45–65, 2003, doi: https://doi.org/10.1016/S0306-4573(02)00021-3.

[6] C. D. Manning, P. Raghavan, and H. Schütze, “Scoring, term weighting, and the vector space model,” in *Introduction to Information Retrieval*, C. D. Manning, H. Schütze, and P. Raghavan, Eds. Cambridge: Cambridge University Press, 2008, pp. 100–123.

[7] S. Robertson, “Understanding inverse document frequency: on theoretical arguments for IDF,” *J. Doc.*, vol. 60, pp. 503–520, 2004, doi: 10.1108/00220410410560582.

[8] C. D. Manning, “Text classification and Naive Bayes,” in *Introduction to Information Retrieval*, C. D. Manning, H. Schütze, and P. Raghavan, Eds. Cambridge: Cambridge University Press, 2008, pp. 234–265.

[9] M. Manjotho, T. Jameel, S. Khanzada, L. A. Thebo, and A. A. Manjotho, “Improving Performance of Mobile SMS Classification Using TF-IDF & Multinational Naive Bayes Classifier,” vol. 2, no. 1, pp. 26–32, 2018.

[10] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[11] P. Refaeilzadeh, L. Tang, and H. Liu, “Cross-Validation,” in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, 2009, pp. 532–538.