Topic Classification of Islamic Question and Answer Using Naïve Bayes and TF-IDF Method

Aura Sukma Andini, Danang Triantoro Murdiansyah*, Kemas Muslim Lhaksmana

School of Computing, Telkom University, Indonesia
**danangtri@telkomuniversity.ac.id*

ABSTRACT

Information spread through the internet is widely used by people. One of the most searched information on the internet is information related to Islamic religious knowledge. However, the large amount of information available from various sources makes it difficult for people to find correct information. In this study, a classification system for Islamic question and answer topics was built using the Naïve Bayes and TF-IDF methods. This study uses 1000 question and answer articles taken from Islamic consultation websites, namely rumahfiqih.com and islamqa.info. The multi-class classification uses five categories which are manually labeled using the category classes on the website. From several test scenarios in this study, the Naïve Bayes classification method using TF-IDF (n-gram level) with a maximum feature of 1000 at data splitting ratio of 70:30 produces the highest accuracy of 81%. The results are compared with the results of the system built using Support Vector Machine.

Keywords: Question and Answer, Topic Classification, Multi-class, Naïve Bayes, TF-IDF.

1. INTRODUCTION

Information spread through the internet is widely used by people to find anything. One of the most searched information on the internet is Islamic religious knowledge [1, 2]. However, the amount of information available from various sources makes it difficult for people to find the correct information. In addition, the head of information about Islamic knowledge must be clear and accurate. Otherwise, the source may be incorrect [2].

This study used the topic classification method for articles containing questions and answers on several Islamic religious consultation websites. The classification is divided into specific topics because each Islamic consulting web has a different classification system [1, 3]. However, the classification based on this topic is done manually by adopting the topic class provided by the Islamic consulting website [4]. Examples include topics of worship, prayer, faith, and so on.

Research related to this has been done, but previous researchers suggested adding website sources and increasing the number of question and answer article data to classify the topic [1]. So that this study was conducted to compare the results of the classification with previous studies. In addition, this question and answer search system is expected to make it easier for people to search for articles, mainly about Islamic knowledge based on the topic.

The existence of many classification methods is a particular concern in this study [4]. Therefore, the author will focus on the classification method used in Naïve

Bayes and using TF-IDF. Previously, with one of the same dataset sources, research using this method already had good results. Research that has been done by Hardifa et al. [1] results in a precision value of 0.88, 0.85 recall, 0.86 f1-score, and 0.97 accuracies. Thus, in this study, the authors developed the number of Islamic question and answer articles data from two different sources and implemented TF-IDF to extract features.

2. MATERIAL AND METHODS

In this study, the researcher built an Islamic question and answer topic classification system those automatically classified topics into five predetermined classes. An overview of the topic classification system for questions and answers about Islam created in this research can be seen in Figure 1.

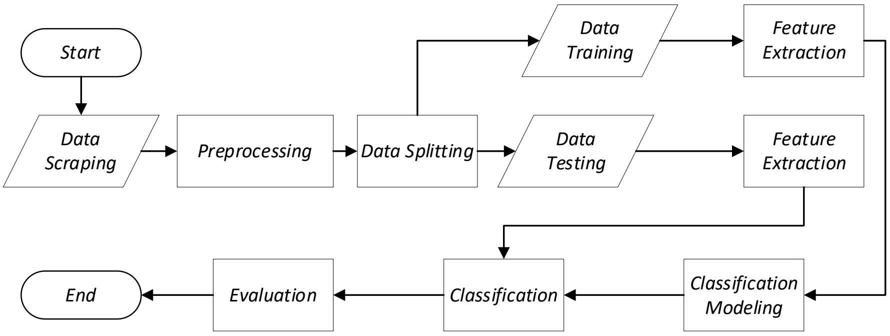

FIGURE 1. System Overview

In this study, researchers collected data on Islamic question and answer articles in Indonesian taken from Islamic consulting websites (rumahfiqih.com and islamqa.info) using web scraping. Web scraping is an important technique used to generate structured data based on unstructured data available on the web [5]. In this study, web scraping was carried out using the "Web Scraper" extension installed on the Google Chrome browser version 60+. The web scraping results in the form of a data collection of pairs of questions and answers are saved in a file with the format (.csv and .xlsx). After the data is collected, a classification system for Islamic question and answer topics is built using Python.

In this study, the data collected amounted to 1000 articles of question and answer pairs. From each of these websites will be taken 500 articles with five different categories. And for each category consists of 100 articles. So, the number of each category is 200 data (data balance). After obtaining the 1000 articles, the multi-class labeling process will be carried out manually. The labels used were adopted from the existing categories on each of the Islamic consulting websites. The five categories used are worship, aqidah, marriage, muamalat, and contemporary. In Table 1 can be seen an example of the results of data scraping that has been given a multi-class label manually.

TABLE 1
Example Dataset

Id	Title	Question	Answer	Class
515	Dana Amal Shalat Jum'at Boleh Digunakan Untuk Apa?	Ass. Ustadz, saya mau bertanya ten tang dana dari amal shalat Jum'at, boleh dipakai untuk apa saja?	Uang kotak amal di masjid yang diisikan oleh para peserta shalat Jumat adalah uang infaq yang bersifat sangat umum.	Muamalah
1218	Benarkah Nabi Isa Punya Ayah?	Bagaimana hukumnya bila ada orang yang meyakini nabi isa mempunyai ayah?	Kalau ada orang yang menyatakan bahwa Nabi Isa \'alaihissam punya ayah, maka orang itu berdusta	Akidah

At the initial stage, the dataset file is entered into the system for pre-processing. This pre-processing process aims to process raw data to become more qualified and efficient when performing the text classification process. In this study, the dataset used is in Indonesian. The pre-processing process uses the help of the Sastrawi library.

The initial stage of doing data pre-processing is data cleaning. At the data cleaning stage, punctuation marks that are deemed unnecessary are removed, such as periods (.), commas (,), exclamation points (!), and others. It also removes redundant numbers and spaces. After the data is clean, the case folding stage is carried out. All letters will be lowercase and non-alphabet characters are also removed. This process helps convert all text in the input document into the standard form [6].

The next stage is the stop word removal process. All unimportant words will be removed. Stop words are words that are not unique in articles or common words that are usually found in documents. For example, conjunctions 'and', 'which', 'or', and so on. After that, the stemming process is carried out, and all words will be converted into essential words and follow the structure of the language used. The last step that is no less important is tokenization. This stage breaks sentences into several tokens or words valuable when modeling text classification [6, 7]. The sample data after the pre-processing process can be seen in Table 2.

TABLE 2
Example of Pre-processing Data Results

Process	Pre-processing Results
Data	Saya telah membaca dalam soal no. 1584, bahwa dalam menetapkan bulan Ramadan cukup penglihatan orang yang adil terpercaya (tsiqah).
Data cleaning	Saya telah membaca dalam soal no bahwa dalam menetapkan bulan Ramadan cukup penglihatan orang yang adil terpercaya tsiqah
Case folding	saya telah membaca dalam soal no bahwa dalam menetapkan bulan ramadan cukup penglihatan orang yang adil terpercaya tsiqah
Stop word removal	membaca no menetapkan ramadan penglihatan adil terpercaya tsiqah
Stemming	baca no tetap ramadan lihat adil percaya tsiqah
Tokenization	[baca, no, tetap, ramadan, lihat, percaya, tsiqah]

After finished doing the pre-processing on the dataset, the next to do process the data splitting. In the data splitting stage, the dataset is separated to be used in the training and testing process. This process can calculate the accuracy of the testing data against the actual label that was done when forming a classification model using training data. As for this study, the data will be divided based on several ratios, namely 90:10 (90% data training, 10% data testing), 80:20 (80% data training, 20% data testing), and 70:30 (70% data training, 30% data testing).

In this study, the label encoding process was carried out on the target variable before dividing the data into training and testing data. Categorical data will be converted into numbers in this process, making it easier for the classification modeling process.

After doing the data splitting, the next step is the feature extraction process. In this process, the text document will be converted into a feature vector, and new features will be created using the existing dataset. Features are the objects whose existence has significant characteristics in the text classification process. The feature extraction process obtains relevant features in the dataset when used in the text classification process [8, 9]. In this study, the feature extraction process will be carried out using the Count Vectorizer and TF-IDF Vectorizer at the word level and n-gram level with the help of the Sklearn library.

Count Vectorizer will build vocabulary from a collection of text documents, then count the occurrences of each word in each document, while TF-IDF has a score that represents the importance/weight of a term in the document and the entire corpus. TF-IDF can be generated at word level and n-gram level. At the word level, it will form a matrix that represents the TF-IDF score for each term in a different document. And at the n-gram level, it will create a matrix representing the TF-IDF score of n-grams (a combination of N terms). The following is the equation for calculating the weights using TF-IDF [8].

$$tf = f_{\mathrm{t,d}} \tag{1}$$

The value of $f_{\mathrm{t,d}}$ in equation (1) is the frequency of occurrence of a term in the document.

$$idf_j = \log\left(\frac{N}{df_j}\right) \tag{2}$$

Based on equation (2), the value of N is the number of documents df_j is the number of documents containing term j. From the *tf* and *idf* values obtained previously, the weight value for a term is calculated, expressed by *w*. Here's the equation to get the value of *w*.

$$w = tf \times idf_j \tag{3}$$

After doing the process of feature extraction, the next step is the process of making a classification model. In this study, the type of Naïve Bayes Classifier used is Multinomial Naïve Bayes because it is suitable for classifying texts or multi-class documents. Multinomial Naïve Bayes can calculate the frequency of each word that appears in each document. The document class is determined from the words that occur and the number of occurrences in each document [10, 11]. Classification using

Multinomial Naïve Bayes has features for each class independently. The following is a formula for text classification using Multinomial Naïve Bayes [12].

$$\hat{P}(c) = \frac{N_c}{N} \tag{4}$$

Equation (4) is a prior probability calculation. N_c is the number of categories c in all documents and N is the number of all documents.

$$\hat{P}(w|c) = \frac{count\ (w,c)+1}{count\ (c)+|V|} \tag{5}$$

Equation (5) is a probability likelihood calculation which is estimated by calculating the number of w words in the category using Laplace (add-1) divided by the total number of words in category c added with $|V|$ which is the number of word variations.

$$c_{MAP} = arg\max_{c} \hat{P}(c) \prod_i \hat{P}(\ x_i|c\ ) \tag{6}$$

Equation (6) calculates the probability of the Maximum a Posterior value of a category.

In addition to the Naive Bayes method, SVM is also one of the methods in supervised learning commonly used for text classification. SVM works to find the best dividing hyperplane between classes by calculating the margin between classes and finding the maximum value. The hyperplane is the dividing line that separates the data between classes, while the support vector is the data that is closest to the hyperplane [13]. SVM is applied to binary classification in the simplest type, which divides data points into positive and negative [14].

After modeling using the Naïve Bayes and SVM classification method, the last stage is the evaluation process using evaluation metrics to see the performance results. In this study, the performance calculation process uses precision, recall, f1-score, and accuracy. (1) Precision is used to measure the exact number of documents by category. (2) Recall is used to measure the number of appropriate documents. (3) F1-score is used to detect the overall result. (4) Accuracy is used to measure the accuracy of all testing data [10]. The following formula is used to calculate the performance.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} \tag{9}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

TP (True Positives) is a correct document and includes all the results of the classification. TN (True Negatives) is a document that is not true but is included in the classification results. FP (False Positives) documents that are correct but not included in the classification results. FN (False Negatives) is a document that is not true and is not included in the classification results.

3. RESULTS AND DISCUSSION

In the first scenario, testing is carried out at the feature extraction process stage. This scenario will be tested against the results of feature extraction using CV and TF-IDF (a word level and the level of the n-gram) by using different parameter max features, namely None, 2000, and 1000. This scenario aims to see the effect of the feature extraction process and the max features parameter on the accuracy value using the Naïve Bayes classification model.

In the second scenario, testing is carried out at the data splitting stage. This test will be used to see the accuracy value using the Naïve Bayes classification method, which is the best in distributing training data and testing data with different ratios. In this scenario, the data is divided into three different ratios.

TABLE 3
Naïve Bayes Accuracy Results (90:10)

Max Features	Accuracy (%)		
	CV	TF-IDF	TF-IDF + N-gram
None	71	73	74
2000	71	**75**	**75**
1000	72	73	74

TABLE 4
Naïve Bayes Accuracy Results (80:20)

Max Features	Accuracy (%)		
	CV	TF-IDF	TF-IDF + N-gram
None	**75**	73	74
2000	74.5	73	73
1000	74	74	73.5

TABLE 5
Naïve Bayes Accuracy Results (70:30)

Max Features	Accuracy (%)		
	CV	TF-IDF	TF-IDF + N-gram
None	79	80.3	80.7
2000	79.3	80.3	79.7
1000	80	80.7	**81**

In the third scenario, testing is carried out at the stage of forming a classification model. In this test, the Support Vector Machine (SVM) classification model will be used as a comparison to see a better accuracy value between the classification models that use Naïve Bayes and SVM. This test is carried out using the same data splitting as in the second scenario.

TABLE 6
SVM Accuracy Results (90:10)

Max Features	Accuracy (%)		
	CV	TF-IDF	TF-IDF + N-gram
None	72	75	75
2000	72	75	75
1000	73	75	**76**

TABLE 7
SVM Accuracy Results (80:20)

Max Features	Accuracy (%)		
	CV	TF-IDF	TF-IDF + N-gram
None	69	76.5	77
2000	69	76.5	**78**
1000	70,5	75.5	76

TABLE 8
SVM Accuracy Results (70:30)

Max Features	Accuracy (%)		
	CV	TF-IDF	TF-IDF + N-gram
None	72.3	78.7	80
2000	72.3	78.7	80.3
1000	72.7	**81**	80

The data will be divided based on several ratios, namely 90:10 (90% data training, 10% data testing), 80:20 (80% data training, 20% data testing), and 70:30 (70% data training, 30% data testing). Based on the test results, in scenario 1, the highest accuracy value is 75%. The highest accuracy results were obtained from TF-IDF (at the word level and n-gram level) with max features of 2000. The results of scenario 1 testing can be seen in Table 3. it turns out that the max features parameter affects the accuracy value. And it can be noticed that the accuracy value using TF-IDF is higher than CV. This could be because the CV only counts the frequency of occurrence of words, while the TF-IDF performs a weighting whose scores represent the importance of a term in the document and the entire corpus. However, it can also be seen that the max features that have been set on CV and TF-IDF respectively have accuracy values that are not much different. This could be due to the similarity of features used in the classification process because initially, CV had 2263 features, TF-IDF (word level) had 2263, and TF-IDF (n-gram level) had 3917 features. If the max features parameter is selected from the initial features to 1000 and 2000, it does not produce a significant accuracy value. Scenario 2 is carried out to strengthen the assumption, which compares if the data is divided by different ratios.

In scenario 2, the highest accuracy value is 81%. The highest accuracy results were obtained from TF-IDF (n-gram level) with max features of 1000 at a data splitting ratio of 70:30. It can be seen in the results of data splitting ratio of 90:10 and 80:20 in Table 3. and Table 4. It has an accuracy value that is not too significant because the highest accuracy result is only 75%. Meanwhile, the results of data splitting with a ratio of 70:30 have a higher value than the ratio of 90:10 and 80:20. This could be due to the increasing proportion of training data that only produces keywords or features that are almost the same and do not add to system knowledge during the classification process. The learning process is sufficient with the features contained in the data, which has a ratio of 70:30. In Table 9. it can be seen the number of initial features used with different data ratios.

TABLE 9
Number of Initial Features

Data Ratio	CV	TF-IDF	TF-IDF + N-gram
90:10	2263	2263	3917
80:20	2273	2273	4033
70:30	2298	2298	4070

In scenario 3, the highest accuracy value from the use of the SVM classification model is 81%. Compared with the results of scenario 2 using the Naïve Bayes classification model, the resulting accuracy values are the same. In scenario 2, the accuracy results from TF-IDF (n-gram level) with a maximum feature of 1000 at a data splitting ratio of 70:30. Meanwhile, in scenario 3, the highest result is TF-IDF (word level) with a maximum feature of 1000 at a data splitting ratio of 70:30.

It can be seen from the comparison of the results of scenarios 2 and 3 in Figures 2, 3, and 4. that the average value of accuracy using Naïve Bayes classification modeling is higher than SVM. This is because SVM is more suitable for use on binary class data than multi-class data. From the comparison results of the Naïve Bayes and SVM classification methods, it can also be seen that the highest average value of accuracy is 80.47%. The highest average value was obtained from the Naïve Bayes classification method at a data splitting ratio of 70:30 and TF-IDF (n-gram level).

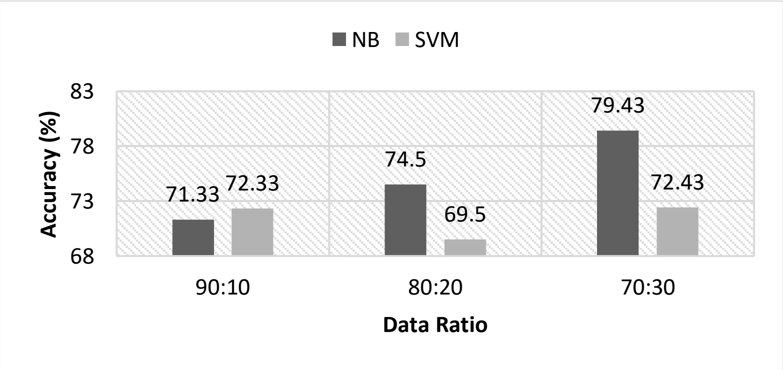

FIGURE 2. Comparison Average Accuracy Results of Naïve Bayes and SVM in CV

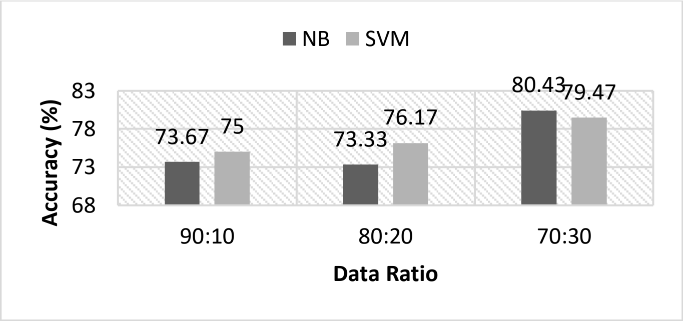

FIGURE 3. Comparison Average Accuracy Results of Naïve Bayes and SVM in TF-IDF

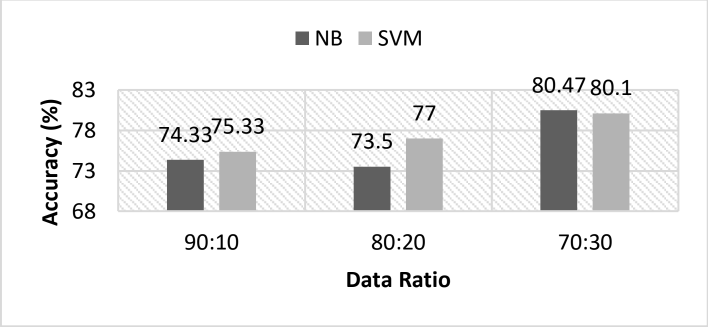

FIGURE 4. Comparison Average Accuracy Results of Naïve Bayes and SVM in TF-IDF + N-gram

4. CONCLUSION

A classification system for Islamic question and answer topics has been built using the Naïve Bayes and TF-IDF methods. Based on the results of testing with several previously carried out scenarios, the Naïve Bayes classification method using TF-IDF (n-gram level) with max features of 1000 at data splitting ratio of 70:30 produces the highest accuracy of 81%. Accuracy value of 81% is also generated by the SVM classification method, but the difference is in SVM the highest accuracy value is using TF-IDF (word level).

From the results, it can also be seen that the Naïve Bayes classification method has higher average accuracy value than SVM. The highest average accuracy value of 80.47% resulted from the Naïve Bayes classification method at data splitting ratio of 70:30 and TF-IDF (n-gram level). Thus, it can be concluded that the effect of the size of the max features on different data splitting ratios can affect the accuracy value. In addition, by using TF-IDF, the system produces better accuracy scores than by not using TF-IDF. And the accuracy of TF-IDF at the n-gram level is better than the word level.

Some suggestions that can be done in further research are increasing the number of Islamic questions and answering article data from several other sites. In addition, it is also recommended to use other classification methods and feature extraction processes with more optimal values than previous studies.

REFERENCES

[1] N. Hardifa, K. Lhaksmana and J. Jondri, "Topic Classification of Islamic Question and Answer Using Naive Bayes Classifier," *Indonesia Journal on Computing (Indo-JC),* 2019.

[2] M. Rahman, N. Samsudin, A. Mustapha and A. Abdullahi, "Comparative Analysis for Topic Classification in Juz Al-Baqarah," *Indones. J. Electr. Eng. Comput. Sci,* 2018.

[3] M. Afandi, A. Adiwijaya and W. Astuti, "Klasifikasi Multilabel Pada Hadis Bukhari Terjemahan Bahasa Indonesia Menggunakan Mutual Information Dan Support Vector Machine," in *eProceedings of Engineering*, 2019.

[4] M. Rahman and Y. Akter, "Topic Classification from Text Using Decision Tree, K-NN and Multinomial Naïve Bayes," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT),* 2019.

[5] A. Saurkar, K. Pathare and S. Gode, "An Overview On Web Scraping Techniques And Tools," *International Journal on Future Revolution in Computer Science & Communication Engineering,* 2018.

[6] J. Han, M. Kamber and J. Pei, Data Mining Concepts and Techniques, San Fransisco, 2012.

[7] Suyanto, Data Mining, Bandung: Penerbit Informatika, 2017.

[8] L. H. Patil and M. Atique, "A novel approach for feature selection method TF-IDF in Document Clustering," in *Proceedings of the 2013 3rd IEEE International Advance Computing Conference*, 2013.

[9] I. Syuriadi, A. Adiwijaya and W. Astuti, "Klasifikasi Teks Multi Label Pada Hadis Dalam Terjemahan Bahasa Indonesia Berdasarkan Anjuran, Larangan Dan Informasi Menggunakan TF-IDF Dan KNN," in *eProceedings of Engineering*, 2019.

[10] S. Putra, Y. Sugiarti, G. Dimas, M. Gunawan, T. Sutabri and A. Suryatno, "Document Classification using Naïve Bayes for Indonesian Translation of the Quran," in *2019 7th International Conference on Cyber and IT Service Management (CITSM),* 2019.

[11] L. Putri, M. Mubarok and A. Adiwijaya, "Sentimen Pada Ulasan Buku Berbahasa Inggris Menggunakan Information Gain Dan Naïve Bayes," in *eProceedings of Engineering*, 2017.

[12] R. Pane, M. Mubarok and N. Huda, "Multi-Lable Classification on Topics of Quranic Verses in English Translation Using Multinomial Naive Bayes," in *2018 6th International Conference on Information and Communication Technology (ICoICT),* 2018.

[13] S. Al Faraby, E. R. R. Jasin and A. Kusumaningrum, "Classification of hadith into positive suggestion, negative suggestion, and information," *Journal of Physics: Conference Series,* 2018.

[14] B. Chen, W. Gu and J.Hu, "An Improved Multi-Label Classification Based on Label Ranking and Delicate Boundary SVM," in *The 2010 International Joint Conference on Neural Networks (IJCNN),* 2010.