INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Question Classification for Helpdesk Support Forum Using Support Vector Machine and Naïve Bayes Algorithm

Noor Aklima Harun
Department Deputy Vice Chancellor (Research and Innovation),
Universiti Teknologi Malaysia
Johor Bahru, Malaysia
Email: aklima@utm.my

Sharin Hazlin Huspi*
Department of Applied Computing and Artificial Intelligence,
Faculty of Computing,
Universiti Teknologi Malaysia
Johor Bahru, Malaysia
Email: sharin@utm.my

Noorminshah A. Iahad
Department of Information Systems,
Faculty of Management,
Universiti Teknologi Malaysia
Johor Bahru, Malaysia
Email: minshah@utm.my

Submitted: 4/10/2022. Revised edition: 13/3/2023. Accepted: 16/3/2023. Published online: 30/5/2023
DOI: https://doi.org/10.11113/ijic.v13n1.388

***Abstract*—The helpdesk support system is now essential in ensuring the journey of support services runs more systematically. One of the elements that contribute to the non-uniformity of the question data in the Helpdesk Support System is the diversity of services and users. Most questions asked in the system are in various forms and sentence styles but usually offer the same meaning making its hard for automation of the question classification process. This has led to problems such as the tickets being forwarded to the wrong resolver group, causing the ticket transfer process to take longer response. The key findings in the exploration results revealed that tickets with a high number of transfer transactions take longer to complete than tickets compared to no transfer transaction. Thus, this research aims to develop an automated question classification model for the Helpdesk Support System by applying supervised machine learning methods: Naïve Bayes (NB) and Support Vector Machine (SVM). The domain will use a readily available dataset from the IT Unit. The results using these techniques are then evaluated using confusion matrix and classification report evaluation, including precision, recall, and F1-Measure measurement. The outcomes showed that the SVM algorithm and TF-IDF feature extraction outperformed in terms of accuracy score compared to the NB algorithm. It is expected that this study will have a significant impact on the productivity of team technical and system owners in dealing with the increasing number of comments, feedback, and complaints presented by end-users.**

***Keywords*—Helpdesk Support, Question Classification, Machine Learning, Support Vector Machine, Naïve Bayes**

I. INTRODUCTION

Support services throughout the ongoing application system development process are crucial and necessary operations for an organization. Technical and management support is essential and critical to ensure the success of an application system, whether, in the user requirement phase, programming development phase, system-testing phase, or final support stage until the system is fully mature. Usually, support services must be continued to ensure the survival of an application system. The helpdesk support system has become a trend for all business in ensuring the journey of support services runs more systematically. Various open-source applications for helpdesk support systems, which are ready to use, have already existed and can be selected by management based on business suitability.

The helpdesk support system usually covers various types of support services and categories of users. This diversity is one of the factors to the non-uniformity of the form of question data. The number of question data will undoubtedly increase in line with the addition of system development. The increment in the number of users will influence the technical and management to answer expeditiously every question asked. Also, the variety of questions that are often asked in the helpdesk support system include questions of various forms and sentence styles but offer the same meaning. If the questions are classified by category, it will aid the accuracy of the solution. Thus, it simplifies the usage of the system while not irritating the process.

Unstructured data is generally difficult to manage, especially in obtaining useful information from the text. The process of text mining is required on unstructured data to support and produce a category of classification of information. This text analysis acquires high-quality information for an extensive collection of documents. Overloading data in line with the increase in the number of systems developed and services provided leads to why the data classification needs to be developed. It ensures an improved, faster organization can provide more economical service continuity.

The problem-solving techniques used in this study will significantly impact the technical team and system owners in dealing with the increasing number of comments, feedback, and complaints presented by end-users. Question classification information is also used to produce interactive dashboards to provide essential insights for system owners and technical teams involved in future decision-making.

II. Literature Review

High-quality customer service is crucial in service and business. Customer satisfaction in receiving the best aid should not be disregarded. Every enterprise's IT service delivery relies on the helpdesk. Many firms utilize intelligent helpdesk solutions to increase customer service quality due to the necessity of high-quality customer service [1].

Apart from the importance of helpdesk support systems that are increasingly emphasized, the ability of helpdesk ticketing systems must also be highlighted. The capabilities of the helpdesk support system include ensuring high availability of services, boosting the helpdesk team's productivity, building the helpdesk that business demands, optimizing asset utilization to ensure maximum Return of Investment (ROI), controlling and managing all IT things, and complementing existing business processes.

For a helpdesk system, one important component is the question and answering (Q&A) system, where it contain question data, which is the leading information to be processed to ensure that Q&A activities can be appropriately implemented. Therefore, the existing questions need to be coordinated to be answered accurately and quickly. A question is a language term used to make information requests [2]. It may categorize inquiries into several sorts based on their information demands, such as what, which, when, and how. Thus, the question type helps organize questions for various information requirements.

In most circumstances, what a consumer wants from NLP is correct answers to inquiries posed by persons [3]. Question classification is an essential part of question processing since it decides the kind of response. The class of answers plays an essential role in Questions Answering Systems since it defines what information must be recovered from a knowledge base [4]. The classification of questions, especially frequently asked questions (FAQ), is to be clustered into the correct categories; therefore, the accuracy of the answers provided is also the primary purpose of this study.

Three essential elements make up a standard Q&A system's architecture: *Question Processing*, *Information Retrieval* and *Answer Processing* [4][5]. A successful Q&A system depends on question classification, the first duty that needs to be passed in the task cycle, as shown in Fig. 1. The following task of information retrieval and answer processing cannot produce sound output to the questioner without proper question classification.

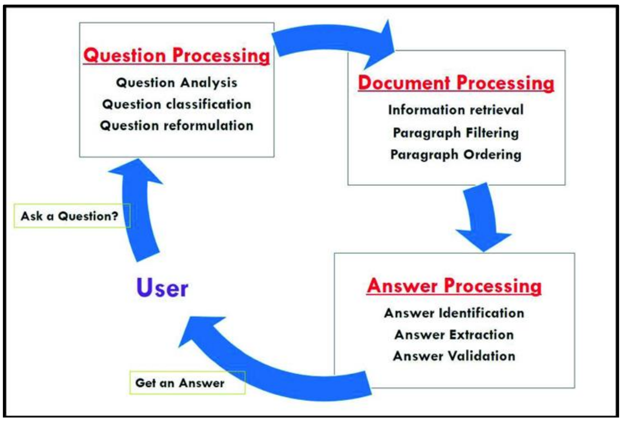

Fig. 1. Q&A Task Cycle

Question Processing is the first element that involves analyzing and classifying questions asked by users. Question analysis, classification, and reformulation are crucial in question processing because they provide helpful information about the expected type of answer. Question classification is a valuable step in question processing because it provides valuable information about the needed type of answer. For a system user who asks the question, "*What should I do because there is no 'Save' button on my system display?*" The technical answer must clearly state through a systematic review the step-by-step that needs to be done for the system's button to be visible to the user. In contrast to questions involving real information such as "*What is the URL for a research proposal application?*" For this, the answer must submit the required URL.

Information Retrieval is also one of the jobs in document processing apart from paragraph filtering and ordering. In [4], they stated that extracting text passages from a set of documents is most likely to contain the answer to the input query.

Answer processing is a method of extracting answer words from a passage of text in order to generate a final response [6]. Answer identification, extraction, and validation are the final process for producing the output sent to the user.

A. Question Classification Approach

Various question classification techniques have been investigated and implemented, especially in customer service-related domains. There are three strategies for question classification tasks: Approaches such as machine learning, rule-based techniques, and hybrid techniques are also available [3].

One study proposed a Support Vector Machine (SVM) classifier that employs bag-of-words (BOW), POS-tag, synonyms, and entity types [9]. This work is similar to the objective of this research, especially in the extraction technique to various features for every given question. However, the dataset used is the Hierarchical Classification Standard. The Corresponding Corpus is a different domain from the target of this investigation, which will focus on the helpdesk support system dataset. While [10] presented a tree-based convolutional neural network (TBCNN) for programming language processing. Structural information is captured by designing a convolution kernel over a program's abstract syntax trees. They used question classification to evaluate the accuracy of TBCNN on the experimental data dataset for question classification from Corpora, and the question data structure studied has similarities with the domain of this study. Labelling the question data is one of the previous research methodologies used in the subsequent study. However, the question classification is not rule-based, which makes it unique and is said to be the first time that neural network modelling beat dedicated human engineering in this task.

Rules-based approaches are used to classify problems. The technology tries to match questions with rules that have been manually written. However, determining the precise criteria necessitates a significant amount of time and effort to comprehend a wide range of question kinds [3]. [11] present a rule-based method for question classification. The first step is it creates a syntactic map using a parse tree. Second, the headword is extracted using possessive unrolling, preposition rolling, and entity identification. Finally, it checks the existence of a pattern that matches the 'wh-word, auxiliary verb, and headword. Once the pattern is found, the question class is returned. This approach is a state-of-the-art approach with a very high dependency level because all the rules are manually created by humans and produce an accuracy rate that is also lofty. Therefore, it is suitable to create a question class more precisely.

Hybrid classification approaches are a concept that utilizes simple classification algorithms. While misclassification instances are typically called noise, they can also provide valuable information for determining the class values of other instances. [12] and [13] both discussed the hybrid classification approaches, where [12] proposed a hybrid method that employs information gain, word similarity, and frequent lexical patterns to avoid using features with a high computational cost. The researchers used default parameters to configure three classifiers: NB, C4.5, and SVM, which are all available in Weka 3.9. The concept of hybrid classification is somewhat complicated due to its high accuracy. However, it is helpful in question classification specific to the Q&A System because it involves more than one layer algorithm that allows for high accuracy output due to multiple rules by each classifier. While in [13] proposed a hybrid approach for Question Classification that employs both syntactic and semantic analysis. It uses dependency relation parsing for syntactic analysis and a WordNet-based feature expansion technique for semantic analysis. The method includes a simple yet effective WordNet-based hypernym expansion mechanism. This study's findings also performed the best comparison, proving their utility for question target categorization tasks.

B. Question Classification in Helpdesk Support Forum

Service agents spend a significant amount of time manually classifying the incoming tickets. With the massive growth of data, the need to automate ticket classification becomes crucial [14]. This part will look at question classification algorithms that have been or are being employed in the helpdesk area, where they conducted common data pre-processing for large-scale support tickets dataset. Then the data is broken down into several classes before word vectorization models are processed for experimental algorithms. Their findings imply that the challenge of classifying IT support complaints can be easily solved utilizing old traditional methods like the Term Frequency — Inverse Document Frequency (TF-IDF) bag-of-words vectorizer with SVM, which lacks the complexity of neural network models. Support Vector Machine (SVM) and Linear Regression (LR) algorithms were utilized as machine-learning models. Standard information retrieval measures such as Precision, Recall, and F-score were chosen for performance evaluation.

Classifying questions is crucial in ensuring that tickets are forwarded to the appropriate support person. For comparison, [15] chose three distinct intelligent text categorization algorithms: the Naive Bayes classifier, the Random Forest classifier, and the Neural Network. According to the researchers, the Naive Bayes method produced somewhat superior outcomes. Naive Bayes, Random Forest, and Neural Network, respectively, have accuracy rates of 74.8%, 72.2%, and 69.4%. Furthermore, compared to other classification algorithms, Naive Bayes is distinguished by simplicity, which helps it better adapt to working with small, low-quality datasets. The question classification of helpdesk tickets should be automated to reduce the time it takes to resolve issues and reduce the number of mistakes during the escalation process. [1] developed iHelp, an intelligent online helpdesk system that uses historical customer–representative interactions to find problems automatically–solution trends. Before the system is built, a case-ranking mechanism is established to aid users in swiftly obtaining answers to new requests. Existing cases must be ranked based on their semantic value to the input request. The case-clustering process is utilized to help customers find answers to their problems; iHelp clusters the top-ranking cases first and then creates a brief synopsis for each case cluster. Finally, request-based case multi-document summarizing is performed to improve the system's usability and provide a quick summary for each case cluster. They discovered that using sentence-level semantic analysis, a mixture language model, and the Sparse Nonnegative Matrix Factorization (SNMF) clustering technique, the helpdesk support system performed better.

The end-user creates a problem ticket in the helpdesk support system by choosing a category and adding a description. Manually picking the ticket category by the end-user may result in tickets being forwarded to the incorrect resolver group. Paramesh and Shreedhara has done two studies on this problem. The first one [16], they suggested to overcome the problems by parsing the unstructured ticket description provided by the helpdesk user, and ML techniques were employed to develop an automated classifier system that auto categorizes the tickets into one of the established categories. Baseline classifiers such as Naive Bayes (NB) and Support Vector Machines (SVM) are utilized, followed by ensemble approaches to generate the classifier models. Ensemble

approaches such as Bagging and Boosting techniques are applied to the basis classifiers for question categorization. The result shows that bagging of individual SVM's called Bagged-SVM classifier outperformed well compared to all other chosen models. The proposed IT question classifier system ensures that tickets are assigned to the appropriate support group, that support resources are effectively utilized, that end-users have a better experience, and that response times are reduced.
The second study [17] they compared four algorithms, namely Multinomial Naïve Bayes, Logistic Regression, K-nearest Neighbour, and Support Vector Machine. This study showed that Support Vector Machine produced the highest level of accuracy of 87% compared to Logistic regression, Multinomial Naïve Bayes, and K-nearest Neighbour, with 81%, 69%, and 67% respectively on the training dataset. Since it worked well for all service desk ticket data samples, the Support Vector Machines (SVM) classifier model was able to reach a reasonable level of accuracy. The suggested automated ticket classifier system improves the end-user experience and customer happiness and the efficient use of support resources, faster ticket resolution times, and business growth.

Another study from [18] utilizes an accurate ticket classification machine learning model to associate a help desk ticket with its correct service from the start, minimizing ticket resolution time, saving human resources, and enhancing user satisfaction. Classification of question data in the helpdesk support system at the university was carried out using four machine learning algorithms. SVM-based machine learning algorithm shows the best model, and experimental results illustrated that classification accuracy for the technique is higher when considering ticket comment, title, and description at the same time. The results show that SVM outperforms three other algorithms, namely J48 (Tree-based), Decision Table (Rule-based), and Naïve Bayes (Bayes-based), based on a higher level of accuracy.

III. Experimental Design

The experiment comprises five primary phases: data preparation, data exploratory, data pre-processing, design modelling and performance evaluation as described in Fig. 2. For each of the phases, different tools and techniques were used to achieve the desired output.

Phase A and Phase B described the initial activities of preparing the dataset and doing exploratory of the dataset. Phase A focused more on the extraction of the data and the purpose of the data exploration is to get a better understanding of the current state of the data. Phase C mainly preparing the question data for the experiment and helping to get initial view of the type of questions being asked. Phase D is the main phase of the experiment, where model development of the classification model is being developed and finally, Phase E is to determine the accuracy of the model developed using evalution tools.

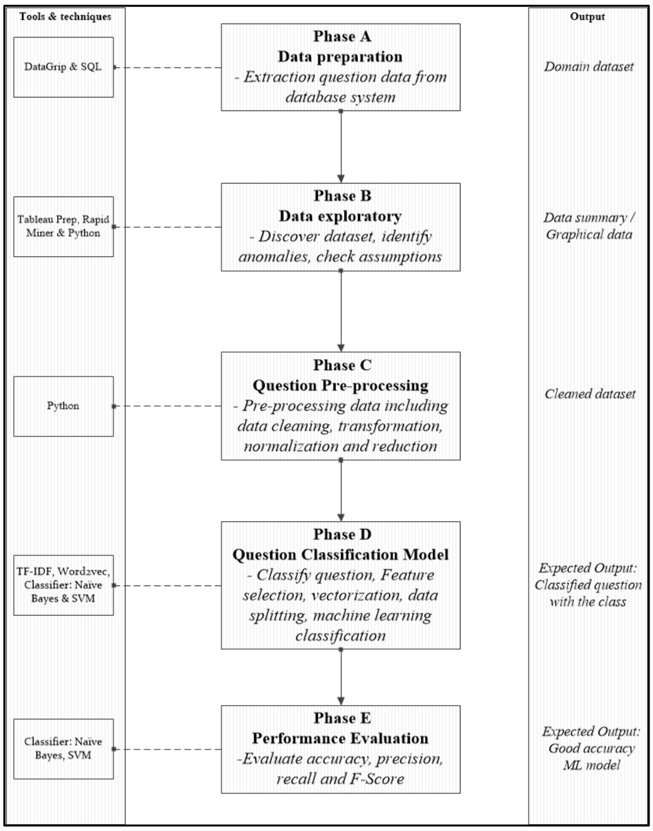

Fig. 2. Question Classification Model framework

A. Dataset and Exploratory Data Analysis (EDA)

The dataset used in this study is the primary data extracted directly from the database that is the entire actual data from the system. It is the dataset from the helpdesk support system, developed and used by the IT Department under the research entity at the University of Higher Learning. The dataset processed and used is of the latest two years' data and involves three categories of users, namely end users, system owners, and system developers. This dataset contains 15 column attributes with 8,404 rows (see Table I).

The exploratory data activity that begins by looking at the detail of values, patents, and possible anomalies or the presence of outliers. The main focus here is to understand the help topic chosen by the users. It was identified that help topics can be classified into three main classes; *Department, Guideline*, and *System* [5]. The *Department* and *Guideline* classes are ticket or question categories that will be sent to the owner of the data in the system, while the *System* class will be answered by the technical system technical.

TABLE I. THE DATASET DESCRIPTION

No.	Attribute	Data Type	Description
1.	thread_entry_id	Numeric	A unique entry id for each transaction in the thread.
2.	thread_id	Numeric	A unique thread id for each transaction in the ticket.
3.	ticket_id	Numeric	A unique ticket id is automatically generated as an special id in the database.
4.	ticket_no	Numeric	The ticket number is automatically generated and as a reference on the end-user display.
5.	poster	String	The user name that generated the transaction in the ticket.
6.	department_category	String	Department-in-charge category for tickets.
7.	category	String	Sub-category under department_category.
8.	title	String	The main title for the ticket generated.
9.	question	String	Descriptions of questions and feedback for tickets generated.
10.	created	Datetime	Created date and time
11.	updated	Datetime	Updated date and time
12.	check state transferred	Numeric	ID for transferred ticket
13.	state transferred	String	Status of transferred ticket

The different class is important and were used to determine the classification of the text forum. Question data that has gone through the first part of pre-processing is labelled manually to the class according to the help topic or question category. Column *target_class* is added in the dataset to store information of classes (System, Department or Guideline). Table II shows the list of classes with the number of Help topic.

TABLE II. NUMBER OF HELP TOPIC FOR EACH CLASS

Class	Number of Help topic
SYSTEM	16
DEPARTMENT	23
GUIDELINE	29

The label determination process for question classification will depends on the question topic as proposed in Table II and will be used in the next phase.

B. *Question Pre-processing*

In this phase, the main data used is the *title* and *question* attribute, where all questions and its answer thread are taken. Example of a question thread is shown in Fig. 3, and it showed an example of ticket transfer due to wrongly chosen help topic by the user.

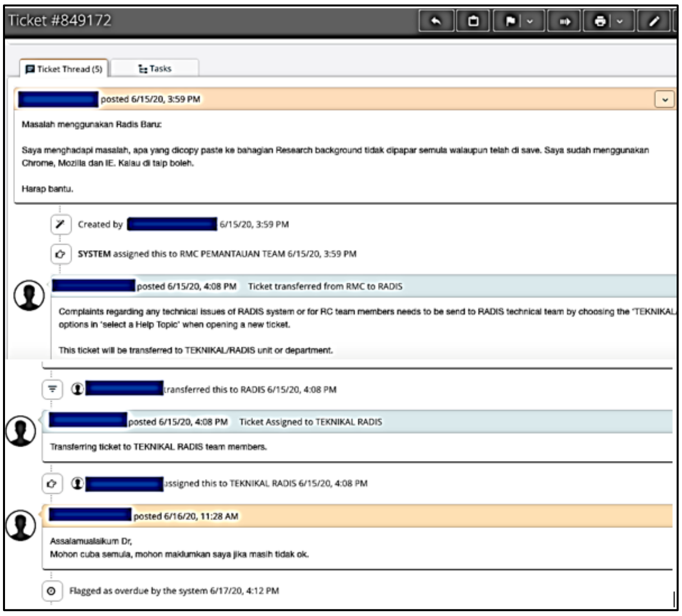

Fig. 3. Example of a ticket thread and ticket transfer process

After the data was extracted from the database, data cleaning process was done using Python and it includes removing all symbols, special characters, punctuations, and short words. Stop words are also identified and discarded and to standardization the data, all question, title, and category are set to lowercase and uppercase according to the suitability of the data.

In the text mining process, data normalization is another important process, which consists of tokenizing, stemming and lemmatizing data is performed to obtain output results that will be easily processed for text classification as in Table III.

TABLE III. THE OUTPUT AFTER DATA NORMALIZATION PROCESS

tokenizedQ	tokenQ_remove_stopwords	pos_tags	wordnet_pos	lemmatized	stem_tokenizedQ
[data, has, been, sent, last, year]	[data, sent, last, year]	[(data, NN), (sent, NN), (last, JJ), (year, NN)]	[(data, n), (sent, n), (last, a), (year, n)]	[data, sent, last, year]	[data, ha, been, sent, last, year]
[application, list, list, action, radis, like, request, list, applications, from, the, sub, menu, list, action, for, resource, sustainability, research, alliance, for, the, role, verifier, umussaa...	[application, list, list, action, radis, like, request, list, applications, sub, menu, list, action, resource, sustainability, research, alliance, role, verifier, umussaa, dah, staff, prof, haslen...	[(application, NN), (list, NN), (list, NN), (action, NN), (radis, NN), (like, IN), (request, NN), (list, NN), (applications, NNS), (sub, VBP), (menu, JJ), (list, NN), (action, NN), (resource, NN),...	[(application, n), (list, n), (list, n), (action, n), (radis, n), (like, n), (request, n), (list, n), (applications, n), (sub, v), (menu, a), (list, n), (action, n), (resource, n), (sustainability...	[application, list, list, action, radis, like, request, list, application, sub, menu, list, action, resource, sustainability, research, alliance, role, verifier, umussaa, dah, staff, prof, haslend...	[applic, list, list, action, radi, like, request, list, applic, from, the, sub, menu, list, action, for, resourc, sustain, research, allianc, for, the, role, verifi, umussaa, dah, staff, and, prof...
[please, done]	[please, done]	[(please, VB), (done, VBN)]	[(please, v), (done, v)]	[please, do]	[pleas, done]
[this, ticket, will, transferred, radis, technical, team]	[ticket, transferred, radis, technical, team]	[(ticket, NN), (transferred, VBD), (radis, JJ), (technical, JJ), (team, NN)]	[(ticket, n), (transferred, v), (radis, a), (technical, a), (team, n)]	[ticket, transfer, radis, technical, team]	[thi, ticket, will, transfer, radi, technic, team]

C. *Question Classification Model*

The development of the model involved feature selection, data splitting, text vectorization or feature extraction, and classification using a machine learning algorithm. The features selection is processed manually and the attributes selected are the question's text and labelled class as in Table II. This process is done manually with the question and target class, and the focus will be given to question data that contain important keywords that formed the basis of the text classification.

The data splitting process is done to meet the requirements of the machine learning process, and here the dataset was split into training and testing data using a suitable ratio. For this study, the summary of the split data is as Table IV.

TABLE IV. SUMMARY OF SPLIT DATASET

	Training Data	**Testing Data**
Percentage	70%	30%
Total samples	1,523	654
Number of sample per target classes	1,263 - System 113 - Department 147 - Guideline	542 - System 49 - Department 63 - Guideline

Feature extraction is the process of extracting a list of words and converting them into a feature set that a classifier will later use. Here text vectorization is applied, where it converts the text words into feature vectors. Text vectorization used the Count Vectorizer (CV) and Term Frequency-Inverse Document Frequency (TF-IDF).

D. *Machine Learning Model Development*

Based on previous studies discussed in the literature review, the Naïve Bayes (NB) and Support Vector Machine (SVM) techniques was selected as it managed to produce higher accuracy for classification. The presence of a given feature or properties in a class is assumed to be independent of any other feature in Naive Bayes [21]. The formula for the Naïve Bayes to calculate the probability is in [22]:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (1)$$

where *x*= attributes and *c*= class.

SVM is a non-probabilistic linear binary classifier used to describe supervised learning models using a separating hyperplane [23]. Due to their primary advantages, such as their robustness in high-dimensional environments, SVM has been employed effectively in many text categorization studies [24]. Based on the stated suitability, both techniques were selected to implement this research.

Another important process done was data resampling was performed for the highest accuracy of feature extraction and algorithm to verify the best result for the algorithm used. It is assumed that the oversampling and under-sampling technique can confirm that the output is the best.

IV. RESULTS AND DISCUSSION

In general, this study developed two key experiments using Naive Bayes (NB) and Support Vector Machine (SVM) algorithms to develop a question classification model. The accuracy, precision, recall, and F1-score were used to generate the classification report for each experiment. The first evaluation was to compare the output for Count vectorizer and TF-IDF vectorizer for both NB and SVM algorithm with different parameter settings from the original dataset.

Based on the first output, a further experiment was conducted by selecting the highest accuracy vectorizer and a machine learning algorithm that was deemed appropriate to proceed with resampling techniques as second evaluation. The dataset was split 70-30 as described in Table IV.

A. *Count Vectorizer (CV) with Naïve Bayes (NB) and Support Vector Machine (SVM)*

Table V and Table VI shows the result from Count vectorizer as features extraction for both NB and SVM. The first experiment was done using Count vectorizer as the features extraction using the default parameter setting (Table V) and Table VI showed result by changing the default parameter setting (for NB, alpha value = 1.8 and SVM, C value = 5.0 and kernel type = RBF).

TABLE V. EVALUATION RESULTS FOR COUNT VECTORIZER USING DEFAULT PARAMETER SETTING

Count Vectorizer	**Naïve Bayes**				**Support Vector Machine**			
Class	**Precision**	**Recall**	**F1-Score**	**Support**	**Precision**	**Recall**	**F1-Score**	**Support**
SYSTEM	0.97	0.85	0.91	614	1.00	0.83	0.91	650
DEPARTMENT	0.31	0.52	0.38	29	0.06	0.75	0.11	4
GUIDELINE	0.08	0.45	0.14	11	0.00	0.00	0.00	0
Weighted Average	0.92	0.83	0.87	654	0.99	0.83	0.90	654
Accuracy score	0.832				0.832			

TABLE VI. EVALUATION RESULTS FOR COUNT VECTORIZER USING NEW PARAMETER SETTING

Count Vectorizer	**Naïve Bayes**				**Support Vector Machine**			
Class	**Precision**	**Recall**	**F1-Score**	**Support**	**Precision**	**Recall**	**F1-Score**	**Support**
SYSTEM	0.98	0.84	0.91	632	0.99	0.84	0.91	636
DEPARTMENT	0.18	0.53	0.27	17	0.18	0.69	0.29	13
GUIDELINE	0.02	0.20	0.03	5	0.03	0.40	0.06	5
Weighted average	0.95	0.83	0.88	654	0.97	0.84	0.89	654
Accuracy score	0.829				0.838			

As shown in Table V, there is no significant difference between NB and SVM. Both results showed accuracy score of 0.832, but SVM failed to identify any questions for the GUIDELINE class. SVM showed slightly better precision score for the SYSTEM class. The result was improved in Table VI, where it showed that accuracy score for SVM was better (0.838) than the default setting and also able to predict questions for the GUIDELINE class. It also showed that the new parameter setting for SVM were able to make question classification more accurately and evenly between classes.

Thus a higher C value has a significant impact on noisy data points and this has helped the SVM hyperplane to prioritize very few misclassifications. It is also observed that lower C value were able to make the hyperplane separate the data points well, but there is a high risk of possible misclassifications. Moreover, changing the kernel type to RBF has an advantage; it can handle the scenario when the relationship between class labels and characteristics is nonlinear [25].

But we can see that changing the parameter did not improved the result for NB. This is expected since increasing the alpha value has the possibility that the model will be bias towards the class which has more records and making the model become a dumb model (underfitting problem). More on resampling the model will discussed later.

B. Term Frequency-Inverse Document Frequency (TF-IDF) with Naïve Bayes (NB) and Support Vector Machine (SVM)

Table VII and Table VIII shows the output of the second performance evaluation using TF-IDF vectorizer as the features extraction. From the tables, NB showed lower accuracy score compared to SVM for both parameter setting based on the result. This might be mainly because NB algorithm only looks independently for each text.

TABLE VII. EVALUATION RESULTS FOR TF-IDF VECTORIZER USING DEFAULT PARAMETER SETTING

TF-IDF vectorizer	Naïve Bayes				Support Vector Machine			
Class	**Precision**	**Recall**	**F1-Score**	**Support**	**Precision**	**Recall**	**F1-Score**	**Support**
SYSTEM	0.86	0.89	0.88	521	1.00	0.83	0.91	653
DEPARTMENT	0.55	0.41	0.47	66	0.02	1.00	0.04	1
GUIDELINE	0.35	0.33	0.34	67	0.00	0.00	0.00	0
Weighted average	0.78	0.79	0.78	654	1.00	0.83	0.91	654
Testing accuracy score	0.787				0.830			

TABLE VIII. EVALUATION RESULTS FOR TF-IDF VECTORIZER USING NEW PARAMETER SETTING

TF-IDF vectorizer	Naïve Bayes				Support Vector Machine			
Class	**Precision**	**Recall**	**F1-Score**	**Support**	**Precision**	**Recall**	**F1-Score**	**Support**
SYSTEM	0.88	0.89	0.89	534	0.99	0.85	0.91	634
DEPARTMENT	0.55	0.44	0.49	61	0.18	0.64	0.29	14
GUIDELINE	0.33	0.36	0.34	59	0.05	0.50	0.09	6
Weighted average	0.80	0.80	0.80	654	0.96	0.84	0.89	543
Training accuracy score	0.803				0.838			

NB's accuracy score improved when the parameter was changed (alpha value was set to 1.8) in Table VIII, but again it might result the model to be underfitted. The SVM model accuracy score also improved when the parameter setting was changed (C value = 5.0 and kernel type = RBF). And the same result as in the previous experiment (using Count vectorizer), where the default parameter did not able to predict the GUIDELINE.

C. Resampling Analysis

Based on the result in the previous experiment, further experiments were done through the data resampling method in order to validate and select the best feature extraction and machine learning algorithm technique. Furthermore, data resampling was done to see if it helped the dataset imbalance, remove bias, and at the same time improve the accuracy score. Here, the number of original samples and the number of samples that change after the resampling process (oversampling or under-sampling) is critical to record so that the effects and impact of resampling will be seen more clearly. As seen in Table V-VIII, SVM showed the highest accuracy in all experiments. Therefore, the resampling experiment will also help to select the best feature vectorizer for SVM.

The first resampling method chosen is the SVM Synthetic Minority Oversampling Technique (SVMSMOTE). SMOTE works by selecting samples in the feature space close together, drawing a line between them in the feature space, then drawing a new sample at a position along that line. In balancing the dataset, SMOTE will look at the minority number of samples and then oversample the data without involving a change in the majority number of the sample.

The resampling method chosen for under-sampling process is the One-Sided Selection (OSS). The method will clean the database by removing noise samples suitable for all machine learning algorithm types. In balancing the dataset, OSS will look at the majority number of samples and then under-sampled the data without involving a change in the minority number of the sample.

The result for the resampling experiments is shown in Fig. 4, and it is shown that SVM algorithm using TF-IDF vectorizer is more suitable to be chosen as the question classification model.

Fig. 4 supported that TF-IDF is the most accurate feature extraction used by SVMs. The result of the SVM algorithm acts more accurately on text classification and interactions between questions. Therefore, question data in the helpdesk is more suitable for using the SVM algorithm as the question classification model because it is crucial to see the interactions between question text since the sentences in the question are usually related.

The experiment using the resampling method also helps validate the best technique to be selected for model development. SVMSMOTE is used for oversampling, and OSS for under-sampling experiments and both supported that SVM + TFIDF combination of techniques seems most suitable for the question classification for the helpdesk support system based on final results in this study.

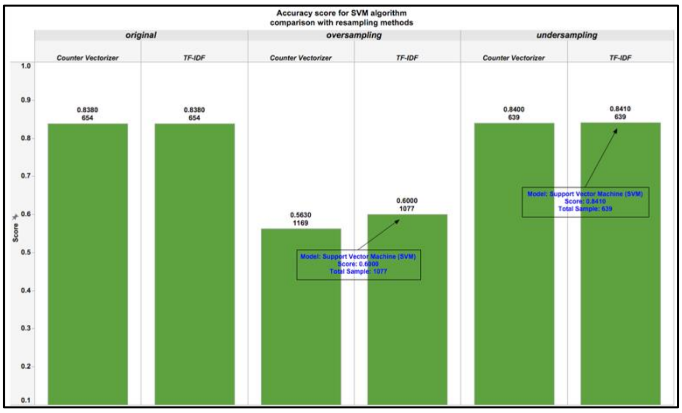

Fig. 4. Accuracy score for NB and SVM algorithm in original, oversampling and under-sampling techniques

V. Conclusion and Future Work

This study shows that the Support Vector Machine (SVM) algorithm develops a better question classification model than the Naïve Bayes (NB) technique for the current dataset.

The first evaluation from the classification experiment was to compare the output for Count vectorizer and TF-IDF vectorizer for NB algorithm and SVM algorithm with different parameter settings from the original dataset. Based on the first output, a further experiment was continued by selecting the highest accuracy vectorizer and a machine learning algorithm deemed appropriate to proceed with resampling techniques as second evaluation The testing dataset is 30% split data which is 645, while the training dataset is 70% split data which is 1,523 (as in the Support column).

One of the limitation for the study is related to the significant imbalance dataset of the proposed classes. This is due to the dataset, which is still new and (only two years of data). Thus, there is the high probability of data overfitting based on the resulting output.

It is suggested that the model development can be improved by continuing research using other appropriate machine learning algorithms, deep learning, or ensemble models. Since the SVM algorithm is more suitable for this research, additional studies, especially the validation part for the predicted question class, must be done thoroughly. The feature extraction process can also be explored using other techniques such as the Word2Vec method.

Acknowledgment

This research was supported by Ministry of Higher Education (MOHE) through *Fundamental Research Grant Scheme* (FRGS/1/2021/ICT02/UTM/02/9). The authors would also like to express the appreciation to the UTM Research Computing (UTMRC) for all the opportunities and facilities provided.

References

[1] Wang, D., Li, T., Zhu, S. and Gong, Y. (2011). IHelp: An intelligent online helpdesk system. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *41*(1), 173-182.

[2] Fan, Z., Wei, Z., Li, P., Lan, Y. and Huang, X. (2018). A question type driven framework to diversify visual question generation. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July, 4048-4054.

[3] Yilmaz, S. and Toklu, S. (2020). A deep learning analysis on question classification task using Word2vec representations. *Neural Computing and Applications*, *32*(7), 2909-2928. Available at: https://doi.org/10.1007/s00521-020-04725-w.

[4] Cortes, E. G., Woloszyn, V., Binder, A., Himmelsbach, T., Barone, D. and Möller, S. (2020). An empirical comparison of question classification methods for question answering systems. *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*. (May), 5408-5416.

[5] N. A. Harun, S. H. Huspi and N. A. Iahad. (2021). Question classification framework for helpdesk ticketing support system using machine learning. *2021 7th International Conference on Research and Innovation in Information Systems (ICRIIS)*, pp. 1-7. Doi: 10.1109/ICRIIS53035.2021.9617077.

[6] Cortes, E.G., Woloszyn, V. and Barone, D.A.C. (2018). *When, Where, Who, What or Why? A Hybrid Model to Question Answering Systems*. Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-99722-3_14.

[7] Hassani, H., Beneki, C., Unger, S., Mazinani, M. T. and Yeganegi, M. R. (2020). Text mining in big data analytics. *Big Data and Cognitive Computing*, *4*(1), 1-34.

[8] Elnagar, A., Al-Debsi, R. and Einea, O. (2020). Arabic text classification using deep learning models. *Information Processing and Management*, *57*(1), 102121. https://doi.org/10.1016/j.ipm.2019.102121.

[9] Xu, S., Cheng, G. and Kong, F. (2017). Research on question classification for automatic question answering. *Proceedings of the 2016 International Conference on Asian Language Processing, IALP 2016*. 218-221.

[10] Mou, L., Peng, H., Li, G., Xu, Y., Zhang, L. and Jin, Z. (2015). Discriminative neural sentence modeling by tree-based convolution. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*. 2315-2325.

[11] Madabushi, H. T. and Lee, M. (2016). High accuracy rule-based question classification using question syntax and semantics. *COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers*. (December), 1220-1230.

[12] Liu, Y., Yi, X., Chen, R., Zhai, Z. and Gu, J. (2018). Feature extraction based on information gain and sequential pattern for English question classification. *IET Software*, *12*(6), 520-526.

[13] Hao, T., Xie, W., Wu, Q., Weng, H. and Qu, Y. (2017). Leveraging question target word features through semantic relation expansion for answer type classification. *Knowledge-Based Systems*, *133*, 43-52.

[14] Wahba, Y., Madhavji, N. H., Steinbacher, J. and Steinbacher, J. (2020). Evaluating the effectiveness of static word embeddings on the classification of IT support tickets. (November).

[15] Beresnev, A. and Gusarova, N. (2020). *Comparison of Intelligent Classification Algorithms for Workplace Learning System in High-Tech Service-Oriented Companies*, Springer International Publishing. Available at: http://dx.doi.org/10.1007/978-3-030-65218-0_27.

[16] Paramesh, S. P. and Shreedhara, K. S. (2019). *Automated IT service desk systems using machine learning techniques*, Springer Singapore. Available at: http://dx.doi.org/10.1007/978-981-13-2514-4_28.

[17] Paramesh, S. P. and Shreedhara, K. S. (2019). IT Help Desk Incident Classification Using Classifier Ensembles. 1980-1987.

[18] Al-Hawari, F. and Barham, H. (2019). A machine learning based help desk system for IT service management. *Journal of King Saud University - Computer and Information Sciences*, *33*(6), 702-718. https://doi.org/10.1016/j.jksuci.2019.04.001.

[19] Sekhar, S. R. M., Siddesh, G. M., Raj, M. and Manvi, S. S. (2021). Protein class prediction based on Count Vectorizer and long short term memory. *International Journal of Information Technology (Singapore)*, *13*(1), 341-348. https://doi.org/10.1007/s41870-020-00528-3.

[20] Lee, S. B., Lim, J. S., Cho, J. S., Oh, S. Y., Whangbo, T. K. and Choi, C. (2020). A classification method of fake news based on ensemble learning. *ACM International Conference Proceeding Series*.

[21] Ray, S. (2021). 6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R. *6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R*. Available at: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ [Accessed June 4, 2021].

[22] Vembandasamy, K., Sasipriya, R. and Deepa, E. (2015). Heart Diseases Detection Using Naive Bayes Algorithm. *International Journal of Innovative Science, Engineering & Technology*, *2*(9), 441-444.

[23] Sueno, H. T., Gerardo, B. D. and Medina, R. P. (2020). Multi-class document classification using support vector machine (SVM) based on improved naïve bayes vectorization technique. *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(3), 3937-3944.

[24] Basari, A. S. H., Hussin, B., Ananta, I. G. P. and Zeniarja, J. (2013). Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization. *Procedia Engineering*, *53*, 453-462. Available at: http://dx.doi.org/10.1016/j.proeng.2013.02.059.

[25] Apostolidis-afentoulis, V. (2015). SVM Classification with Linear and RBF kernels. *ResearchGate*. (July), 0-7.