EAI Endorsed Transactions
on Scalable Information Systems Research Article **EAI.EU**

A Chatbot Intent Classifier for Supporting High School Students

Suha K. Assayed[1,*], Khaled Shaalan[1], and Manar Alkhatib[1]

[1]Faculty of Engineering and Information Technology, British University in Dubai, UAE

Abstract

INTRODUCTION: An intent classification is a challenged task in Natural Language Processing (NLP) as we are asking the machine to understand our language by categorizing the users' requests. As a result, the intent classification plays an essential role in having a chatbot conversation that understand students' requests.
OBJECTIVES: In this study, we developed a novel chatbot called "HSchatbot" for predicting the intent classifications from high school students' enquiries. Evidently, students in high schools are the most concerned among all students about their future; thus, in this stage they need an instant support in order to prepare them to take the right decision for their career choice.
METHODS: The authors in this study used the Multinomial Naive-Bayes and Random Forest classifiers for predicting the students' enquiries, which in turn improved the performance of the classifiers by using the feature's extractions.
RESULTS: The results show that the random forest classifier performed better than Multinomial Naive-Bayes since the performance of this model is checked by using different metrics like accuracy, precision, recall and F1 score. Moreover, all showed high accuracy scores exceeding 90% in all metrics. However, the accuracy of Multinomial Naive-Bayes classifier performed much better when using CountVectorizers compared to using the TF-IDF.
CONCLUSION: In the future work, the results will be analysed and investigated in order to figure out the main factors that affect the performance of Multinomial Naive-Bayes classifier, as well as evaluating the model with using a large corpus of students' questions and enquiries.

Keywords: intent classification, features extraction, countvectorizer, tf-idf, multinomial naive-bayes, random forest, chatbot, nlp

Received on 10 September 2022, accepted on 31 October 2022, published on 21 December 2022

doi: 10.4108/eetsis.v10i2.2948

1. Introduction

Nowadays, students from different stages in schools enjoy using different digital communication channels during their education as it can increase their learning experiences. For instance, some schools use emails, Google Classroom, Seesaw, ClassDojo, Zoom, video calls and other digital platforms. However, in this era of time, technology is rapidly improving and becoming exceedingly fast paced, hence, schools and colleges should deploy the latest technology in order to empower students to utilize it. Apparently, chatbot is considered as one of the most advanced technologies that can communicate and interact between machines and humans and it has recently been applied in both secondary education and higher education in many aspects, for instance, some universities use a chatbot and AI for answering the students' questions that are related to particular courses [1]. Indeed, high school is one of the most crucial stages in students' lives, as in this stage they can shape their future career with their passions and interests. Thus, it could be a stressful time for students especially when they have concerns about their standardised exams, college majors, universities and vocational streams [2]. Unsurprisingly, schools' advisers and counsellors are the main source of higher education and career information for both students and their families

* Corresponding author. Email: sassayed@gmail.com

[3]. High schools, also known as, secondary schools are the most important stage in a student's life as it takes part in shaping their careers, and it starts from grade 9 to grade 12. These four stages of school in the U.S. are called (Freshman, Sophomore, Junior, Senior).

Indeed, students in this stage need an instant support from their schools, but not all schools have enough advisers to all students. Besides, advisors and coordinators at schools are also benefited of using the chatbot by avoiding the repetitive questions and answers [1]. Therefore, in this paper a novel chabot will be developed to be an effective and affordable virtual academic adviser to assist students in high schools by classifying their enquiries and questions to unique labels and categories. This technique is called an intent classification and it will be used mainly as high schools' conversational assistant, despite the fact that this stage is the most important stage in students' life, the target audience in most studies are students in universities and colleges. Hence, integrating this technology in educational platforms and applications would be benefited the high school students

1.1. Intent Classification

Intent classification is one of the biggest challenges of using Natural Language Processing (NLP), as we are asking the machines to understand languages [4] by categorizing users' request/ enquires based on their intentions. As a result, the intent classification plays an essential role in having a chatbot conversation.
In this study, we are asking the chatbot to understand students' request by categorizing their requests to auto labels based on their goals such as Scholarship, HS Curriculum, Study Majors, etc.

2. Related Work

2.1. Chatbot for Intent Classification

Many researchers are inspired to study the chatbots as a conversational assistants and for classifying the intent of users' input. For instance, Henfy et al. [5] conducted a study for developing an intent classifier of chat messages that were used in communicating between the teams of software developers. Additionally, Schuurmans & F. Frasincar [6] used several Machine Language algorithms for improving the intent classification of dialogue utterances. Furthermore, Pérez-Vera et al. [7] developed a chatbot classifier to answer to the users' FAQ by using 5000 tweets from the twitter's account of one of the electricity's companies as it can enhance their services, accordingly increasing the customers' satisfaction.
In fact, researchers have developed chatbots for assisting users in different purposes, such as responding to different enquires about information security [8], or assisting patients in health care services [9] and others.

However, in the next section, the authors focus on chatbots that are applied into educational institutions, particularly in high schools and universities.

2.2. Chatbot for Education

Most researchers in educational fields developed chatbots to learn subjects such as languages, and sciences. Sarosa et al. [10] integrated a chatbot into Facebook for helping students to learn English in Telkom Vocational High School Malang. However, another chatbot called "Ellie" was developed by other researchers to motivate students in Korean classes to learn English by having a chatbot as an English conversation partner, which resulted in a positive impact in encouraging students to engage in conversation [11]. In addition to that, a nother chatbot was designed to improve the learning of Spanish at the National University of Distance Education in Spain, as well other chatbots for learning Chinese [12, 13]. Nevertheless, some studies highlighted the advantages of learning independently by using the chatbot compared to face-to face school interactions [14, 15, 16].

2.2.1 Chatbot for Advising Students

School advisers and counsellors are the main resources of universities and career information for both students and their families [3]. The high school or secondary school are the most important stage in students' life in shaping their careers. According to American School Counsellors Association, the main task of Academic counsellors is helping students to succeed and assisting them to plan their career effectively [17]. In addition to the important role of academic advisors in high schools, simultaneously, a few researchers have developed a chatbot for guiding and advising students toward their future careers. In fact, most public and private high schools around the world have academic counsellors or academic advisors, and this position can be exhibited in different names such as academic adviser, student adviser, academic counsellor, career adviser, etc. Moreover, it is agreed that they have a crucial role in assisting students to get admitted into the right majors and colleges that can fit their interests and personalities [3] and accordingly it can influence students' success at colleges and universities, as studies show that students with more academic readiness during their high schools show higher persistence rate at universities [18]. Furthermore, some universities accept students for "undecided majors" particularly for students who are still not sure about their future careers, and academic advisors can work closely with them to help them select their interested majors. Sanata et al. [1] proposed a chatbot called S.A.N.D.R.A to assist new students from a Brazilian Public University to enquire for any courses by classifying their questions intentions, as well as to answer the most FAQ that students might need during their freshmen year. Moreover, Zahour et al. [2] used a DialogueFlow tool for developing a chatbot for guiding the undergraduate and graduate students to enter the job market by applying the

John Holland theory for determining the dominant type of personality. Similarly, Abbas et al. [19] established a chat platform for encouraging the online engagement between students during the academic year 2021/2022 at the University of Leeds, and the results show a positive impact in students' reactions, as freshman students are more comfortable asking questions due to needing more guidance during the first year. In addition, some colleges adopted a chatbot for relieving stress and improving the student life by using a chatbot counselling system, which can provide different psychological services through Q&A system [20]. Yet, most researchers focused on chatbots for assisting freshman students at colleges, even though many students need to decide their majors in grade 11 and 12 and they should be aware of standardised exams as well as universities earlier. Moreover, some international scholarships required students to declare their majors before applying to the scholarship as these agencies sponsor students based on their interested majors, for example, STEM majors might be accepted for one scholarship and other majors might not be. Hence, this proposed project will focus on providing a chatbot for classifying and labelling the intention of high school (HS) students' enquiries. This proposed chatbot will be called a "HSchatbot" and it will receive different questions from different HS students, for instance, it could be enquiries about scholarships, university requirements, majors, curriculums, studying in the UK, or in the US, and etc. Then the HSchatbot will classify the intention of their requests based on words that are used in their questions, and as a result this chatbot will extract the right meaning from different enquiries. This HSchatbot will be an intent classifier for processing the students' inputs by selecting the most accurate model. In this study different ML algorithms will be executed by using particular features as selection techniques in order to improve the accuracy of the classifier. Figure 1 shows the general framework of HSchatbot, and figure 2 shows the diagram of processing the intent classifier in the HSchatbot.

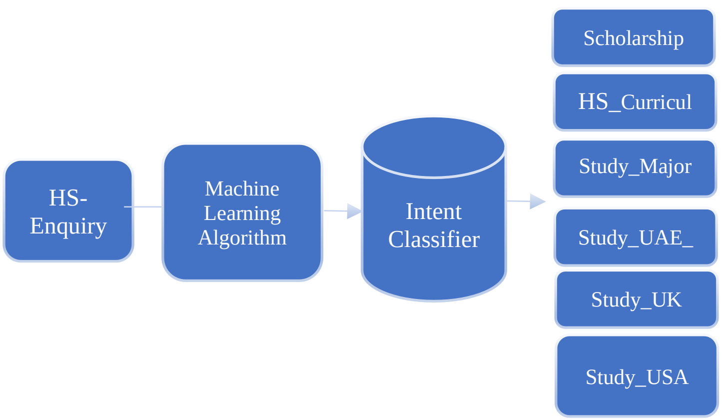


Figure 1. The Framework of the HSchatbot intent classifier

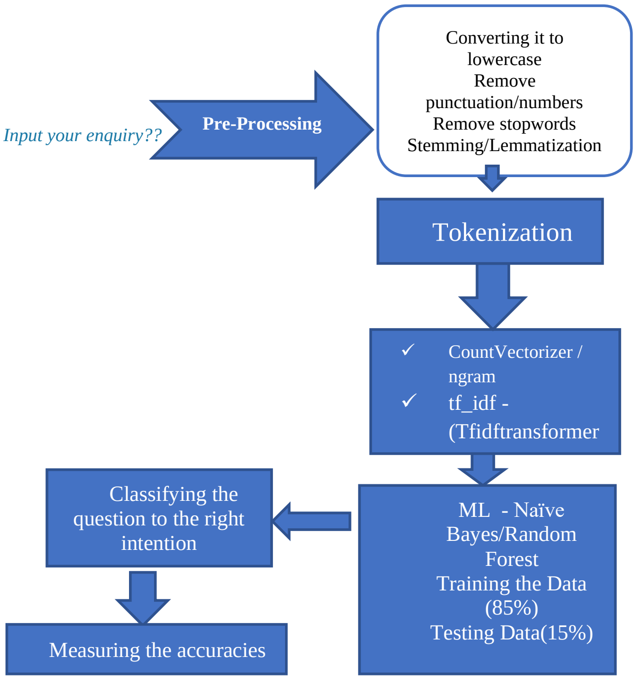


Figure 2. The diagram of processing the HSchatbot intent classifier

3 Experiment and Development

3.1. Enquiries Dataset

The dataset in this study is collected from different academic advisors in schools as well as from different enquiries in academic institutions', where 505 enquiries were collected based on advisors' experiences and others were collected from universities' websites. The data are labeled manually based on the subject of each question as shown in table 1.

Table 1. The intent classification of the dataset.

Ser.	Label/ Intent classification
0	Scholarship
1	HS_Curriculum
2	Study_Majors
3	Study_UAE_Univ
4	Study_UK_Univ
5	Study_USA_Univ
6	Univ_Ranking

The dataset is classified based on the categories (tags), and these tags are assigned to the different enquires. As shown in figure 3 the "Study Majors" and "Study_USA_Univ" tags are mapped to the most of students' questions.

3.2. Preprocessing

Pre-processing is an essential step for any application of Natural Language Processing System (NLP) since all elements in the texts are necessary to be inspected and analyzed through the entire stages of the text processing [21], the purpose of this task is making the text more readable for machine learning algorithms. The main prep-processing functions are tokenization, lower casing, stop words removal and lemmatization.

3.3. Features Extraction

```
[123]: x= df_dataset["Intent_Name"].value_counts()

[124]: print(x)

Study_USA_Univ     129
Study_Majors       127
Study_UAE_Univ      75
Study_UK_Univ       64
HS_Curriculum       43
Scholarship         34
Univ_Ranking        33
Name: Intent_Name, dtype: int64
```

Figure 3. The classification of the dataset.

The Features Extraction for retrieving information quantifies the features by converting the data from unstructured text to structural data which computers and machine language can identify and process effectively [22]. The authors in this article used the features extraction techniques of CountVectorizer & TfidfTransformer in order to improve the performance of the classifier.

3.3.1 CountVectorizer

It is a Scikit-learn package for tokenizing the texts (breaking down a sentence/question) into tokens and selecting the words (features) that occur most frequently. In other words, the CountVectorizer is calculating the FT or the Bag of Words (BoW), simultaneously, it could enhance the performance of this function by including it with the ngram_range argument option which can expand the words from single words to more than one word phrases, figure 4 & figure 5 show the shape of the vector before and after using the pre-processing functions. Figure 4 shows the sparse matrix with 1195 words (features) before applying the pre-processing functions.

```
[142]: word_count_vector.shape

[142]: (505, 1195)
```

Figure 4. The shape of the vector before applying the pre-processing functions.

On the other hand, figure 2 shows 951 words only as several pre-processing techniques have been applied into the 505 sentences (students' enquiries) such as lower casing, stop words removal and lemmatization.

```
[120]: word_count_vector.shape

[120]: (505, 951)
```

Figure 5. The shape of the vector after applying the pre-processing functions

As a result, a matrix of the pre-processed data is created by transforming each question/enquiry into scalar vectors, and each word in the sentence defines as a column in the below matrix and each question or enquiry is shown as an arrow in the matrix as represented in table 2. Moreover, all words are encoded with unique numbers which can be understood by machine language. Figure 6, shows the code that is used in encoding the words into numbers.

Table 2. An explanation matrix of the results of CountVectorizer.

All Features(Vocabs)	0	5	6	….	951
	Id 406	get 360	uk 842		
Question1	0	0	1		
Question2	0	0	0		
Question3	0	1	0		
Question4					
……					
Question 505					

```
[24]:
# Printing the identified Unique words along with their indices
print("Vocabulary: ", cv.vocabulary_)

# Encode the Document
word_count_vector = cv.transform(df_dataset["Comment"])

# Summarizing the Encoded Texts
print("Encoded Document is:")
print(word_count_vector.toarray())
```

```
Vocabulary:  {'id': 406, 'like': 486, 'apply': 62, 'abroad': 4, 'scholarship': 716, 'get': 3
60, 'uk': 842, 'stay': 775, 'safe': 711, 'study': 783, 'find': 332, 'internal': 435, 'colleg
e': 160, 'grant': 372, 'offer': 576, 'consider': 175, 'demonstrate': 220, 'financial': 331,
'need': 560, 'determine': 225, 'sport': 761, 'good': 365, 'way': 871, 'pay': 605, 'merit': 5
```

Figure 6. The Python code of vectorising the texts

After encoding each particular word, the countVectorizer basically creates a matrix with documents or with (Bag-of Words). In simple words, it counts the tokens/ words in each sentence (question) as shown in figure 7.

```
Encoded Document is:
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 1]]
```

Figure 7. The Python results of CountVectorizer after encoding and counting the words (Sparse Matrix).

3.3.2 TfidfTransformer

This function is used for weighing the tokens (words) that occur in the majority of documents (questions and enquiries) in order to weigh again the count of features for all sentences, which in turn reduces the impact of more frequent words with the less valuable words, enhancing the impact of infrequent and more valuable words [23]. That is the reason that TfidfTransformer returns a float number, compared to CountVectorizer which returns an integer number due to only counting the number of times that word comes into the document. Consequently, this function is very effective since it's not focusing only on the frequencies of the words, but it rather focuses on the importance of each particular word.

In fact, this function is calculated by using both the (Term Frequency) TF and (Inverse Document Frequency) IDF. The TF calculates the number of times that a particular feature (word) is repeated in the enquiry/question, as it indicates the importance of this particular word in this particular question (sentence), so if more words were repeated in the same sentence, it indicates the importance of this word. In contrast, the IDF calculates the number of enquiries (sentences) that have this particular word, and IDF will exhibit a lower number if more enquiries and questions have this feature. However, by multiplying (TF * IDF) the results with high scores will indicate the most significant words in the entire sentences

3.4. Machine Learning Algorithms

3.4.1. Multinomial Naive-Bayes

Naive Bayes (NB) is a one of the probabilistic algorithms that depends on the probability theory and Bayes' Theorem to predict the tag/label of the sentence and it calculates the probability of each feature based on previous knowledge of data that are related to this particular feature. NB assumes that all features are independent and equally important, which could sometimes cause a poor performance due to it being an unrealistic assumption. One can improve the performance and reduce the impact of this assumption by using the feature extraction and selection [24].

In this study, the authors use the Multinomial Naive-Bayes, since there are multiple features and each feature represents word count (frequency) and as a result, this kind of algorithm will work very well with these discrete features.

Afterward, the feature vectors that are received from TfidfTransformer will arrive as input to the Multinomial Naive-Bayes classifier.

3.4.2. Random Forest Classifier

The RF is a collection of decision trees, and each decision tree is comprised of sampling data from the training set, and it combines the output of various decision trees in order to reach the particular result. [25].

Random Forest is considered one of the most accurate classification algorithms, as the performance metric for the classification accuracy is high compared to other classifiers [26].

3.5 Testing the Classifier Model

The testing phase is very important to validate the model. The unseen data from students' enquiries (without labels) were tested in Multinomial Naive Bayes and the Random Forest classifiers and both were measured by executing the testing metrics: Precision, Recall and F score. Moreover, the author used feature techniques such as CountVectorizers & TF-IDF in order to enhance the performance of this model.

4 Results and Discussion

Multinomial Naive-Bayes and Random Forest classifiers are used in this study for finding the best classification of students' intention based on their enquiries' content. Students' questions can vary between high schools' curriculum to the majors and universities in particular countries. However, while collecting the data, the students are asked to mention the country in their university's enquiry. Moreover, the countries that are mainly in the corpus are the US, UK and UAE. Nonetheless, the authors used some of the feature extraction techniques such as the CountVectorizers & TF-IDF

4.2 The performance with CountVectorizers

While using the CountVectorizers, the model of Random Forest Classifier is shown to be more accurate than the Multinomial Naive-Bayes classifier as the Accuracy score reached (91%) compared to (88%) in Multinomial Naive-Bayes, as shown in table 3 and figure 8.

Table 3. The performance of the model with using CountVectorizers.

Classifier with TF-IDF	Accuracy	Precision	Recall	F1-Score
Multinomial Naive-Bayes	83%	81%	83%	81%
Random Forest Classifier	92%	93%	92%	92%

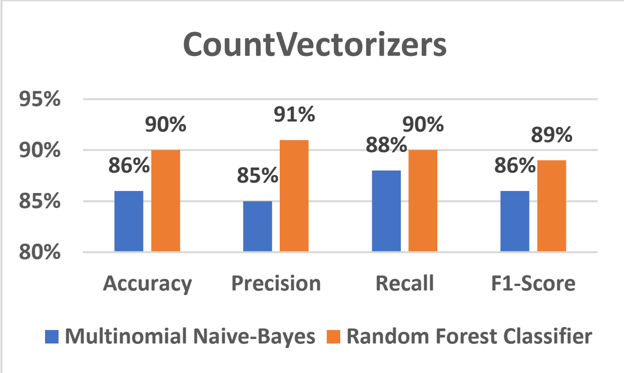


Figure 8. The performance metrics of using the CountVectorizers feature in our study.

4.2 The performance with TF-IDF

The results after using the TF-IDF feature also show that Random Forest Classifier is more accurate than Multinomial Naive-Bayes classifier as shown in table 4 and figure 9.

Table 4. The performance of the model with TF-IDF

Classifier with CountVectorizers	Accurac y	Precisio n	Recal l	F1-Scor e
Multinomial Naive-Bayes	86%	85%	88%	86%
Random Forest Classifier	90%	91%	90%	89%

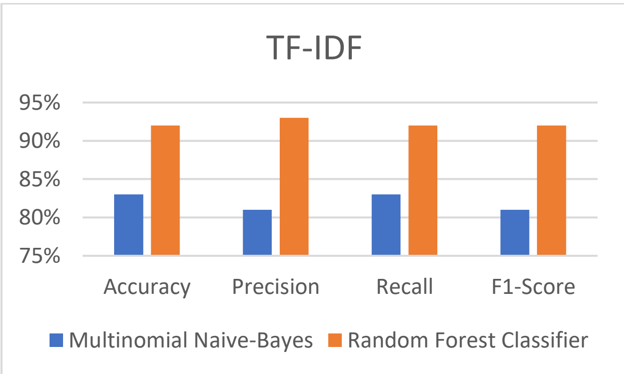


Figure 9. The performance metrics with the TF-IDF

Interestingly, the performance of Multinomial Nave-Bayes was much better while using the CountVectorizers compared with using the TF-IDF as shown in table 5. For more clarification, the performance in both classifiers are compared separately in figure 10 and figure 11

Table 5. The performance of the metrics before and after using the feature selection (TF-IDF)

Classier Name		Accuracy	Precision	Recall	F1-Score
Multinomial Naive-Bayes	TF-IDF	83%	81%	83%	81%
	Count Vectorizers (TF)	86%	85%	88%	86%
Random Forest Classifier	TF-IDF	92%	93%	92%	92%
	Count Vectorizers (TF)	90%	91%	90%	89%

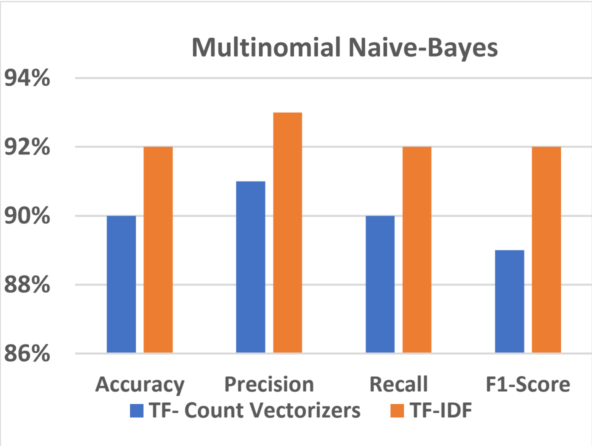


Figure 10. The difference in performance between CountVectorizers and the TF-IDF in the Multinomial Naive-Bayes

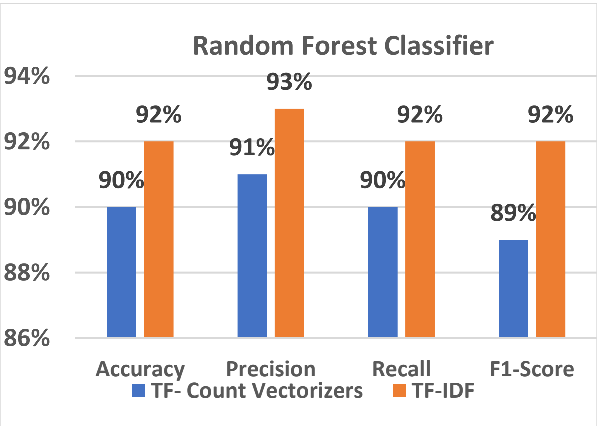


Figure 11. The difference in performance between using the TF and TF-IDF in the RF classifier.

More investigations and studies should be done in order to clarify and understand the factors behind these results. For example, it could be due to the limited size of the corpus. Another reason could be repetition of several terms in students' enquiries as we asked the students to mention the country of the university i.e. UAE, UK, and USA, so these words could be repeated in most of their enquiries and the results of the CountvVectorizers depends on the frequency of the terms in the corpus, therefore the results might be impacted differently.

5 Conclusions and Future Work

Senior students at high schools have higher stress levels regarding their future compared to other students, particularly in selecting the right majors and universities that fit with their interests. Therefore, these concerned students need answers to their enquiries effectively. In this study a HSchatbot is developed in order to provide students with 24/7 support services. This chatbot can understand their questions by categorizing their enquiries' intention to the prober classification. Moreover, in this study the authors used the Multinomial Naive-Bayes and Random Forest classifiers while using the feature extractions (CountVectorizers and TF-IDF) for improving the performance. However, the performance shows more improvement in using the Random Forest classifiers with all feature techniques as it achieved high scores in all metrics, mainly by using the TF-IDF with (93%) in precision score. On the other hand, the Multinomial Naive-Bayes achieved high score by using the CountVectorizers (TF) compared to TF-IDF, hence why this result needs to be investigated more by selecting a large number of corpora as used with other classifiers. This study will be a resource for researchers who are interested in using ML & AI for classifying the students' intentions toward their career. In future work, we will have a large corpus of enquiries without having to ask students to mention the country that are related to their questions, such as "MIT university- in USA "" or Cambridge university in UK", etc. This will be the case to avoid increasing the frequencies of some particular terms. In addition to that, in the future we will extend this study by including deep learning algorithms in order to improve the performance of our model.

Appendix A. Evaluating the performance with CountVectorizer -Python code

A.1. Naïve Bayes Classifier

```
[22]: from sklearn.naive_bayes import MultinomialNB
      from sklearn.ensemble import RandomForestClassifier
      X_train, X_test, y_train, y_test = train_test_split(df_dataset['Comment'], df_dataset['Intent

[42]: count_vect = CountVectorizer()
      X_train_counts = count_vect.fit_transform(X_train)
      #tfidf_transformer = TfidfTransformer()
      #X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
      clfmodel  = RandomForestClassifier(n_estimators=500)
      clf = MultinomialNB().fit(X_train_counts, y_train)
      #clf=clfmodel.fit(X_train_counts, y_train)

[43]: test_data_features = count_vect.transform(X_test)
      #X_new_tfidf = tfidf_transformer.fit_transform(test_data_features)
      X_test_counts = test_data_features
      y_predicted = clf.predict(X_test_counts)

[44]: from sklearn.metrics import accuracy_score
      from sklearn.metrics import accuracy_score
      accuracy = accuracy_score(y_test, y_predicted)

[45]: print(accuracy)

      0.8627450980392157

[46]: from sklearn.metrics import precision_score, recall_score, f1_score
      f1_score(y_test, y_predicted, average='weighted')

[46]: 0.856282618310482
```

A.2. Random Forest Classifier

```
[22]: from sklearn.naive_bayes import MultinomialNB
      from sklearn.ensemble import RandomForestClassifier
      X_train, X_test, y_train, y_test = train_test_split(df_dataset['Comment'], df_dataset['Intent

[52]: count_vect = CountVectorizer()
      X_train_counts = count_vect.fit_transform(X_train)
      #tfidf_transformer = TfidfTransformer()
      #X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
      clfmodel  = RandomForestClassifier(n_estimators=500)
      #clf = MultinomialNB().fit(X_train_counts, y_train)
      clf=clfmodel.fit(X_train_counts, y_train)

[53]: test_data_features = count_vect.transform(X_test)
      #X_new_tfidf = tfidf_transformer.fit_transform(test_data_features)
      X_test_counts = test_data_features
      y_predicted = clf.predict(X_test_counts)

[54]: from sklearn.metrics import accuracy_score
      from sklearn.metrics import accuracy_score
      accuracy = accuracy_score(y_test, y_predicted)

[55]: print(accuracy)

      0.9019607843137255

[56]: from sklearn.metrics import precision_score, recall_score, f1_score
      f1_score(y_test, y_predicted, average='weighted')

[56]: 0.8931751267789132
```

Appendix B. Evaluating the performance with TfIdf features- Python code

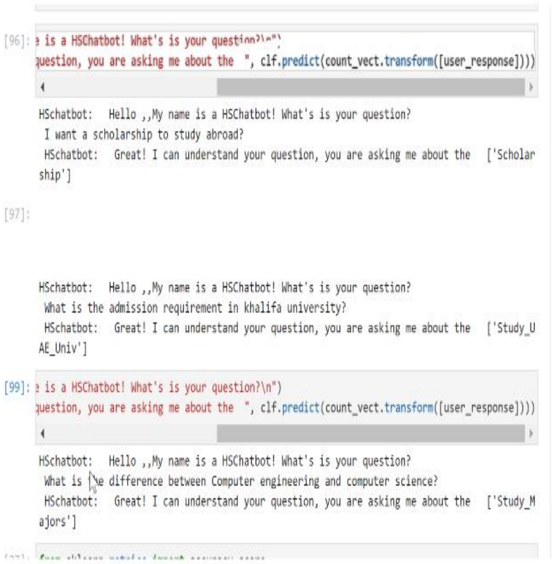
B.1. Naïve Bayes Classifier

```
[168]: count_vect = CountVectorizer()
       X_train_counts = count_vect.fit_transform(X_train)
       tfidf_transformer = TfidfTransformer()
       X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
       clf = MultinomialNB().fit(X_train_tfidf, y_train)
       #clf = RandomForestClassifier().fit(X_train_tfidf, y_train)

[99]:  Decoding_label = label_encoder.inverse_transform(df_dataset["Intent_Name"])
       #print (intent_name_label.unique()), df_dataset["Intent_Name"].unique()

[169]: #testing the accuracy of the naive bayes classifier
       test_data_features = count_vect.transform(X_test)
       X_new_tfidf = tfidf_transformer.fit_transform(test_data_features)
       y_predicted = clf.predict(X_new_tfidf)
       from sklearn.metrics import accuracy_score
       #accuracyscore = accuracy_score(y_test,y_predicted)

[170]: from sklearn.metrics import accuracy_score
       accuracy = accuracy_score(y_test, y_predicted)

[171]: print(accuracy)

       0.8289473684210527

[172]: from sklearn.metrics import precision_score, recall_score, f1_score
       f1_score(y_test, y_predicted, average='weighted')

[172]: 0.8100232198142416

[173]: precision_score(y_test, y_predicted, average='weighted')
```

B.2. Random Forest Classifier

```
[161]: count_vect = CountVectorizer()
       X_train_counts = count_vect.fit_transform(X_train)
       tfidf_transformer = TfidfTransformer()
       X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
       #clf = MultinomialNB().fit(X_train_tfidf, y_train)
       clf = RandomForestClassifier().fit(X_train_tfidf, y_train)

[99]:  Decoding_label = label_encoder.inverse_transform(df_dataset["Intent_Name"])
       #print (intent_name_label.unique()), df_dataset["Intent_Name"].unique()

[162]: #testing the accuracy of the naive bayes classifier
       test_data_features = count_vect.transform(X_test)
       X_new_tfidf = tfidf_transformer.fit_transform(test_data_features)
       y_predicted = clf.predict(X_new_tfidf)
       from sklearn.metrics import accuracy_score
       #accuracyscore = accuracy_score(y_test,y_predicted)

[163]: from sklearn.metrics import accuracy_score
       accuracy = accuracy_score(y_test, y_predicted)

[164]: print(accuracy)

       0.9210526315789473

[165]: from sklearn.metrics import precision_score, recall_score, f1_score
       f1_score(y_test, y_predicted, average='weighted')

[165]: 0.9153629225161554

[166]: precision_score(y_test, y_predicted, average='weighted')

[166]: 0.9302318295739348
```

Appendix C. Testing the classifier



References

[1] Santana, R., Ferreira, S., Rolim, V., de Miranda, P. B., Nascimento, A. C., & Mello, R. F. (2021). A Chatbot to Support Basic Students Questions. In *LALA* (pp. 58-67).

[2] Zahour, O., El Habib Benlahmar, A. E., Ouchra, H., & Hourrane, O. (2020). Towards a Chatbot for educational and vocational guidance in Morocco: Chatbot E-Orientation. *International Journal*, *9*(2).

[3] Cranmore, J., Adams-Johnson, S. D., Wiley, J., & Holloway, A. (2019). Advising high school students for admission to college fine arts programs. *Journal of School Counseling*, *[17]*(10).

[4] Alonso, P. (2020). *Faster and More Resource-Efficient Intent Classification* (Doctoral dissertation, Luleå University of Technology).

[5] Hefny, A. H., Dafoulas, G. A., & Ismail, M. A. (2020, December). Intent classification for a management conversational assistant. In *2020 15th International Conference on Computer Engineering and Systems (ICCES)* (pp. 1-6). IEEE.

[6] J. Schuurmans and F. Frasincar, "Intent Classification for Dialogue Utterances," in IEEE Intelligent Systems, vol. 35, no. 1, pp. 82-88, 1 Jan.-Feb. 2020, doi: 10.1109/MIS.2019.2954966.

[7] Pérez-Vera, S., Alfaro, R., Allende-Cid, H. (2017). Intent Classification of Social Media Texts with Machine Learning for Customer Service Improvement. In: Meiselwitz, G. (eds) Social Computing and Social Media. Applications and Analytics. SCSM 2017. Lecture Notes in Computer Science(), vol 10283. Springer, Cham. https://doi.org/10.1007/978-3-319-58562-8_21

[8] Hamad, S., & Yeferny, T. (2020). A chatbot for information security. *arXiv preprint arXiv:2012.00826*.

[9] Shinde, N. V., Akhade, A., Bagad, P., Bhavsar, H., Wagh, S. K., & Kamble, A. (2021, June). Healthcare Chatbot System using Artificial Intelligence. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 1-8). IEEE

[10] Sarosa, M., Kusumawardani, M., Suyono, A., & Wijaya, M. H. (2020). Developing a social media-based Chatbot for English learning. In *IOP Conference Series: Materials Science and Engineering* (Vol. 732, No. 1, p. 012074). IOP Publishing.

[11] Yang, H., Kim, H., Lee, J. H., & Shin, D. (2022). Implementation of an AI chatbot as an English conversation partner in EFL speaking classes. *ReCALL*, 1-17.

[12] Vázquez-Cano, E., Mengual-Andrés, S., & López-Meneses, E. (2021). Chatbot to improve learning punctuation in Spanish and to enhance open and flexible learning environments. *International Journal of Educational Technology in Higher Education*, *18*(1), 1-20.

[13] Chen, H.-L., Vicki Widarso, G., & Sutrisno, H. (2020). A chatbot for learning chinese: learning achievement and technology acceptance. *Journal of Educational Computing Research*, *58*(6), 1161–1189.

[14] Yin, J., Goh, T. T., Yang, B., & Xiaobin, Y. (2021). Conversation technology with micro-learning: The impact of chatbot-based learning on students' learning motivation and performance. *Journal of Educational Computing Research*, *59*(1), 154-177.

[15] Tseng, J.-J. (2018). Exploring tpack-sla interface: insights from the computer-enhanced classroom. *Computer Assisted Language Learning*, *31*(4), 390–412.

[16] Fryer, L. K., Nakao, K., & Thompson, A. (2019). Chatbot learning partners: connecting learning experiences, interest and competence. *Computers in Human Behavior*, *93*, 279–289. https://doi.org/10.1016/j.chb.2018.12.023

[17] American School Counselor Association (2022). School Counselor and Roles & Ratios. Retrieved from https://www.schoolcounselor.org/About-School-Counseling/School-Counselor-Roles-Ratios

[18] Oripova, M. The Impact of Intrusive College Academic Advising on High School Students' College Degree Attainment Commitment Levels: A Quantitative Quasi-Experimental Study. *Available at SSRN 4076232*.

[19] Abbas, N., Whitfield, J., Atwell, E., Bowman, H., Pickard, T., & Walker, A. (2022). Online chat and chatbots to enhance mature student engagement in higher education. *International Journal of Lifelong Education*, 1-19.

[20] Lin, A. P., Trappey, C. V., Luan, C. C., Trappey, A. J., & Tu, K. L. (2021). A Test Platform for Managing School Stress Using a Virtual Reality Group Chatbot Counseling System. *Applied Sciences*, *11*(19), 9071

[21] Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., Nithya, M., Kannan, S., & Gurusamy, V. (2014). Preprocessing techniques for text mining. *International Journal of Computer Science & Communication Networks*, *5*(1), 7-16.

[22] Q. Liu, J. Wang, D. Zhang, Y. Yang and N. Wang, "Text Features Extraction based on TF-IDF Associating Semantic," *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, 2018, pp. 2338-2343, doi: 10.1109/CompComm.2018.8780663.

[23] Zhao, G., Liu, Y., Zhang, W., & Wang, Y. (2018, January). TFIDF based feature words extraction and topic modeling for short text. In *Proceedings of the 2018 2Nd International Conference on Management Engineering, Software Engineering and Service Sciences* (pp. 188-191).

[24] Shaban, W. M., Rabie, A. H., Saleh, A. I., & Abo-Elsoud, M. A. (2021). Accurate detection of COVID-19 patients

based on distance biased Naïve Bayes (DBNB) classification strategy. *Pattern Recognition*, *119*, 108110.
[25] IBM Cloud Education (2020, December,7). *Random Forest* https://www.ibm.com/cloud/learn/random-forest
[26] Lemons, K., 2020. A Comparison Between Naïve Bayes and Random Forest to Predict Breast Cancer. *International Journal of Undergraduate Research and Creative Activities*, 12(1), pp.1–5. DOI: http://doi.org/10.7710/2168-0620.0287