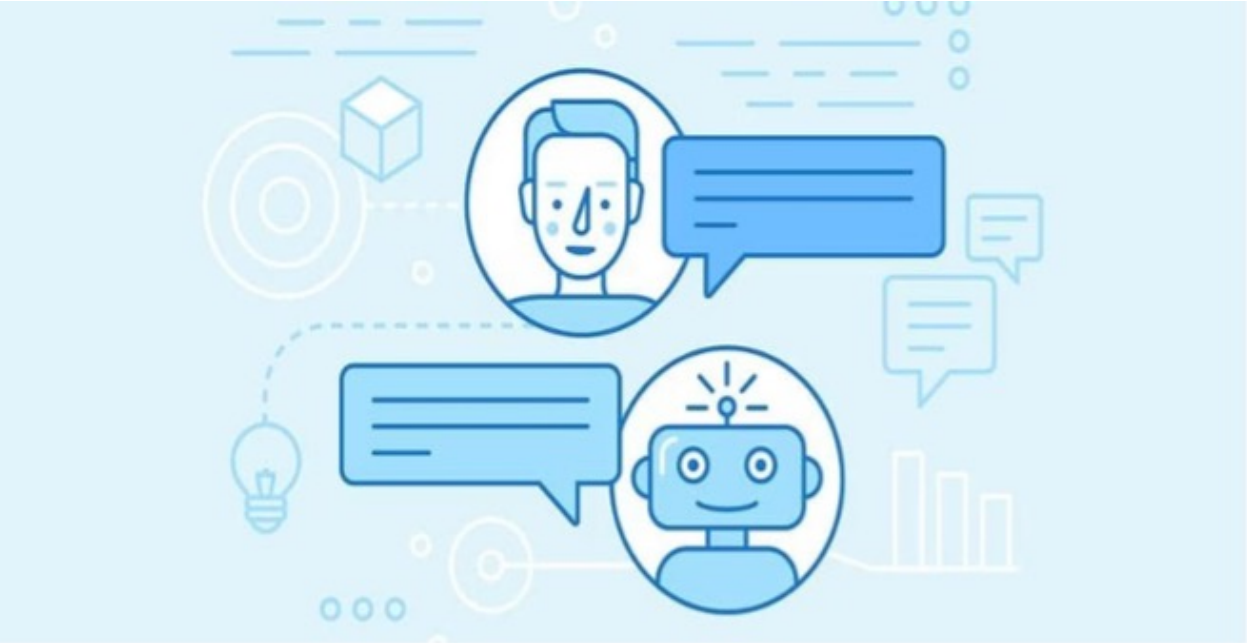
Improving University Customer Support with Learning to Rank Chatbot



Han Chau – 20012654

Manvendra Shrimal - 20011277

Yashita Vajpayee - 20011237

Contents

1 Introduction:

The aim of this research is to develop a chatbot for Stevens University that can efficiently handle a wide range of student support inquiries. The chatbot will be designed to understand natural language queries and provide accurate and relevant information to users. We will also use algorithm to rank or decide the answer's as per the question. By creating a well-designed and well-trained chatbot, we aim to enhance the overall student experience, increase efficiency, and improve accessibility to information.

2 Background and relative work:

Dahiya (2017) defined a chatbot as a computer program that simulates communication and returns a response learned from a trained database using Boolean models to select an exact match for the user's input through pattern matching. Likewise, Shivam et al. (2018) introduced a chatbot model that utilizes a pattern matching technique to extract information from the user's query and compare it with the knowledge base to provide the suitable response. [5]

These methods are simplicity and often quite precise because it retrieves only responses that explicitly contain all of the query terms. However, it also has some notable drawbacks, such as a lack of ranking capability and sensitivity to noisy data or variations in query syntax.

In 2020, Wagh et al. demonstrated the methods used in their system were machine learning, natural language processing, pattern matching, and data processing algorithms. To select a response to a given input statement, they used "MultiLogicAdapter". One issue that can arise is when multiple logic adapters provide the same statement, resulting in an incorrect response.

Our proposed approach involves using learning to rank to get most related answer based on the relevance of the query. By using this technique in combination with natural language processing and pattern matching, we hope to improve the overall effectiveness of the chatbot and provide a better user experience. While learning to rank is still a developing field, we believe it holds great potential for enhancing the capabilities of chatbots and other natural language processing applications.

3 Objectives:

The main aim of this project is to design and develop a chatbot that is tailored to university-related inquiries. The chatbot will be equipped to efficiently handle a broad spectrum of inquiries, ranging from academic program details and admission procedures to campus facilities and student life.

To gather the necessary data, web-scraping techniques will be employed to extract information from the university's website.

In order to train the chatbot to predict the ranking of answers for each user query, we will employ the 'Learning to rank- Pointwise' approach, utilizing the SVM and XGBoost algorithms.

The chatbot will be designed to recognize and respond to variations in user queries, including differences in wording and syntax. This feature will ensure that users can receive accurate and relevant information even if their queries are not formulated in an exact manner.

4 Data Collection:

- We scrape the questions and answers data from different FAQ section from Stevens' website for e.g. Housing and Dining, OPT, New Student Orientation, Interlibrary Loan and Document Delivery Services, etc.
- All these website are hosted on different pages so we create function to scrape it. As some webpages are different than others for e.g. Workday and Disability services we created more function to get data from different HTML class.

- Overall, our sample size is 332 Q&A using Web-scraping and 203 Q&A pairs were obtained from the School of Business. We will use the scraped data to make Model 1 and latter data for Model 2 as it has a query index column.

Some insights into the function used for Scraping:

- We utilized 5 functions to scrape questions and answers from the FAQ section of Stevens' website.
  1. **getFAQ()**: We can extract the questions and answers from each individual div tag using its class name. This method is commonly used in most of the FAQ links we have collected.
  2. **getFAQ1()**: Both the questions and answers are contained within a single div tag. The div tag has children that are h4, and h3 tags, which correspond to the questions, while the other children correspond to the answers.
  3. **getFAQ2()**: This function is similar to getFAQ1() in that the questions and answers are included in one div tag, but in this case, the h4, h3, and h2 tags corresponding questions.
  4. **getFAQ3()**: One of the links has a div tag containing children h2 and p tags. The questions and answers are included in the children of p tags. However, in addition to these answers, there is also an answer contained in the ol tags.
  5. **getFAQ4()**: there is one link in which the children of div tag are h2 and p tags. The first p tag is just the introducing statement and the h2 tags are the statement, we need to remove the text of these tag. In the other p tags, each p tag contain each paragraph, so there are some answers may include more one paragraph. To organize the datasets, we have added a new column called "Category" that contains the category of the questions and answers. Then, we saved the scraped data into a CSV file named "Dataset_Scraping.csv".

- To organize the datasets, we have added a new column called "Category" that contains the category of the questions and answers. Then, we saved the scraped data into a CSV file named "Dataset_Scraping.csv".
  - To avoid confusion, additional information about the problem is provided as many questions do not display distinguishing features.
  - For example:
    * "How do I make a first appointment?" - This question relatives "Counseling and Psychological Services CAPS" category. If there is no category, we can not know which field students want to ask.
    * "What is a Peer Leader?" - "New Student Orientation" category etc.

Category	Questions	Answers
Housing and Dining	When will students find out about their housing assignments	Returning students (all students other than new, first-year students) who select a bed space during the room sele
Housing and Dining	How long are the beds? What kind of sheets will I need?	All the beds in the residence halls are xl twin beds (36 inches wide by 80 inches long). You will need extra long (8
Housing and Dining	Are pets allowed?	No. With the exception of documented service animals, pets are not permitted in Stevens Housing.
Housing and Dining	Can I bunk my bed?	Yes, our beds are designed to be bunked. If you need assistance bunking your bed, please contact your Resider
Housing and Dining	Can furniture be removed from the room and replaced with n	No, the furniture provided may not be removed from the rooms. Non-Stevens furniture is not permitted.
Housing and Dining	How do I get back into my room if I am locked out and do no	Contact the Harries Tower front desk to report a lockout. We will then have the lock core changed. You and your
Housing and Dining	What does a student do if they have a roommate problems?	We encourage all students to complete the online roommate agreement within two weeks of a new assignment.
Housing and Dining	What does a student do if they do not like their room assignr	Students should first reach out to their RA to share their concerns. The RA will assist with problem-solving if the
Housing and Dining	Can I have an overnight guest?	Residents must obtain their roommates' permission before having a guest. Residential students may only have a
Housing and Dining	How often are rooms cleaned?	Resident rooms are cleaned before arrival. Students are responsible for cleaning their own rooms throughout the
Housing and Dining	Where can students do their laundry when living on campus?	Each on-campus residence hall has free laundry with washers and dryers in the building. Stevens Leased Housir
Housing and Dining	Are students allowed to have a refrigerator and microwave in	Yes; however it is highly suggested that you do not purchase your own. We recommend that you rent a MicroFric
Housing and Dining	How does the meal plan work?	Stevens offers meal plans for residential and commuter students. Plans are assigned by class year and building
Housing and Dining	What are DuckBills?	DuckBills are an add on to the meal plan. They are a cash equivalent that can be utilized at any of our food or ve
Housing and Dining	Can money be added to my Duck Bill account?	Yes. You can add cash to the VTS machine in the library or visit theOnline Card Office to use a debit/credit card.
Housing and Dining	Is the food plan available during holidays and breaks?	There is no food service during Thanksgiving and Winter breaks. There may be no meal plan service or reduced s

- In addition, we have utilized the 'FAQ Watson Dataset' provided by the School of Business at Stevens in our project. This dataset contains a comprehensive collection of question-answer pairs, along with their respective categories and sub-categories. We utilized the question,answer and Category column of this dataset.This dataset has proved to be a valuable resource in training and evaluating the performance of our chatbot.

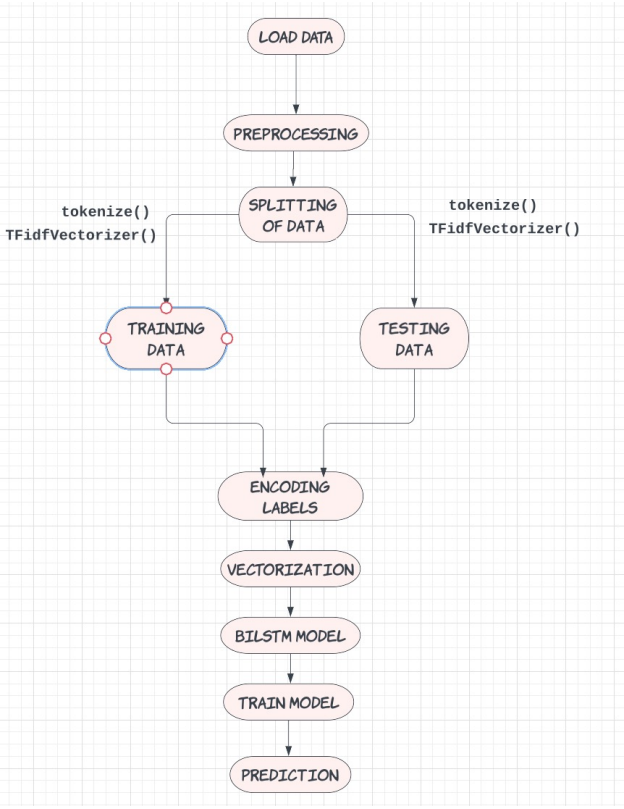
5 Data pre-processing:

Data Augmentation:

- To enhance the size of our dataset and improve the accuracy of our model, we employed data augmentation techniques. Specifically, we utilized methods such as synonym replacements and random insertions to generate additional variations of the existing question-answer pairs.
- Furthermore, we combined questions and answers from different categories and introduced a label column to categorize non-related pairs as 0 and original related question-answer pairs as 1. This approach helped with the supervised learning process of our models.
- Overall, data augmentation proved to be a beneficial approach in enhancing the size and diversity of our dataset, which in turn improved the performance of our chatbot.
- We created a new dataset to predict if the question and answer are paired, where each question has three answers: one answer with a corresponding label of 1 and two randomly selected answers with corresponding label of 0.
- We utilize the LabelEncoder() method to convert the categorical values in the "Category" feature into numerical values. This is done to avoid any potential confusion as some questions may contain similar words in the category's name but not necessarily have the same meaning.
- We randomly selected 80% of the samples for training and reserved the remaining 20% for testing.
- We create some functions:
  - The function **tokenize()** is used to split the question and answer into individual words. It allows for different options to be set, such as lemmatizing, removing stop words, creating bigrams, and removing words that are less than 2 letters.
  - The function **compute_tfidf()** is used to calculate the smoothed tf-idf score for each word in the corpus. Alternatively, we can use TfidfVectorizer() from Scikit Learn.
  - The function **assess_similarity()** is used to calculate the cosine similarity between the question and answer, which is a measure of how similar they are.
- We will preprocess "FAQS WATSON.xlsx" `Question` and `Answer` by:
  - Removing special charachters
  - Tokenization
  - Lemmatization

6 Methodology:

6.1 Training to predicting Category:

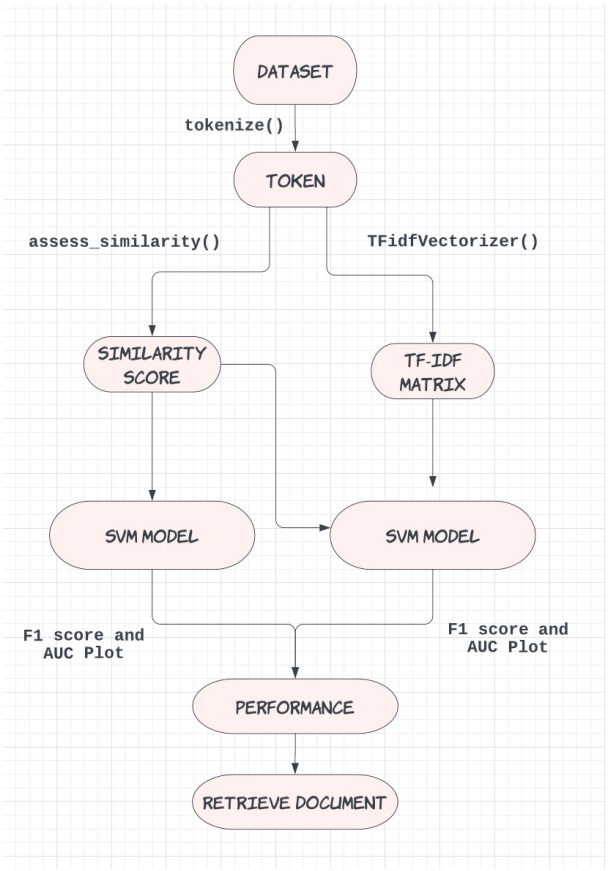


- We split our data into training and testing sets using the train_test_split function, with a test size of 20%. We tokenized the questions using the tokenize function, with options for lemmatization, stopword removal, and bigram creation. We then used TfidfVectorizer to convert the text data into numerical vectors, which were used as inputs to our machine learning models.
- We trained and evaluated several machine learning models on our dataset, including Decision Tree, Random Forest, Logistic Regression, and XGBoost. According to our model evaluation results, the Random Forest model had the highest accuracy with an average accuracy of 99.69% and a standard deviation of 0.34%. The XGBoost model followed with an average accuracy of 99.33% and a standard deviation of 0.40%. The Decision Tree model and Logistic Regression model also performed well with average accuracies of 98.87% and 98.92%, respectively.
- To evaluate the effectiveness of our university chatbot, we conducted testing using both pre-trained questions and new questions. Pre-trained questions were sourced from the 'Watson Dataset', which we utilized for training our chatbot. These questions cover a wide range of topics related to university life, including academic programs, admissions, student life, and campus facilities.
- In addition to pre-trained questions, we also tested our chatbot with new questions that were not present in the training dataset. This allowed us to assess the model's ability to recognize and respond to variations in questions, as well as its generalizability to new inputs.

6.2 Retrieve document problem:

6.2.1 Model 1 - SVM:

- To implement this model, we follow the below graph:

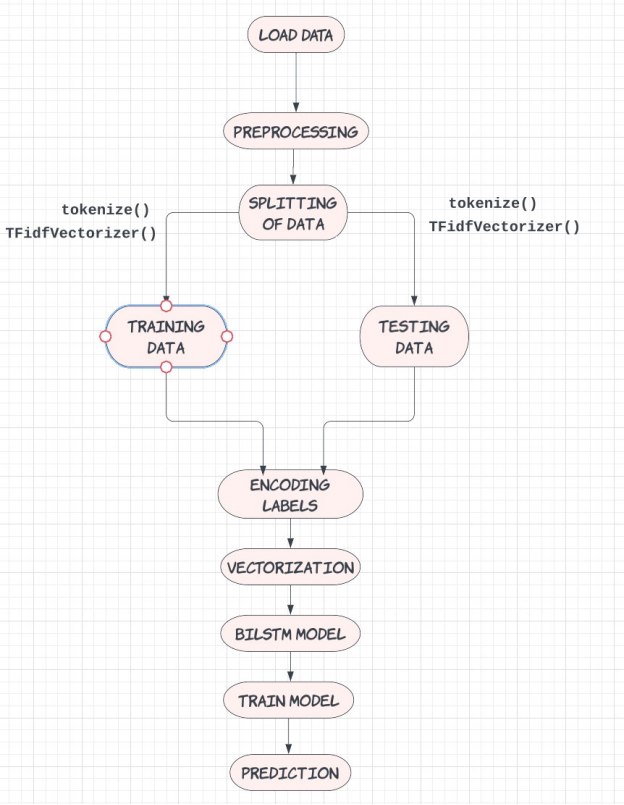


- The dataset to train the SVM model includes 3 columns: Questions, Answers and Label.
- To create the input for the model, we implement several steps:
  - We use tokenize() function to extract the unigram (more than 2 letters) or bigram token. The unigram still keep font after remove punctual such as “I-20”, “can’t”, or “Steven’s”.
  - We calculate the cosine similarity between the questions and answers from assess_similarity() function.
  - We also find the tf-idf matrix by TfidfVectorizer() from scikit-learn.
- We have two inputs for this model:
  - Input 1: just use the cosine similarity.
  - Input 2: combine the tf-idf matrix and cosine similarity with two cases (remove and no remove stop-words).
- We evaluate the performance of the model by classification_report() function and plot the AUC and PRC.
- Then, we test the performance when retrieve the top documents for each question.
  - Questions to test are taken from the ChatGPT paraphrased from the collected questions.
  - Picking randomly 100 questions from this dataset and create the new dataset with 4 columns: Category, Questions Collect, Question Chat GPT, and Answers.

	Category	Question Collect	Question Chat GPT	Answers
0	CPT	What is the difference between CPT and OPT?	What are the differences between CPT and OPT?	In short, CPT is employment that is directly r...
1	Seeking Help Off-Campus	Is it really private and secure?	How secure and confidential is the service?	Yes, we are HIPAA and FERPA compliant which me...
2	Seeking Help Off-Campus	Can I continue working with my counselor once ...	Is it possible to continue working with my cou...	Yes, you will have the option to continue work...
3	CPT	My internship ended early. What documents shou...	How should I report an early end date for my i...	If your CPT employment is ending or ended prio...
4	Innovation Expo	Is registration required to attend?	Is registration mandatory for attending the Expo?	Registration is NOT required, but setting up l...

- Calculating the accuracy of the retrieved documents in the top-1, top-3, and top-5 documents for 2 situations (remove and no remove stop-words).
- Visualizing the top-3 answers for a query.

6.2.2 Model 2 - BiLSTM:



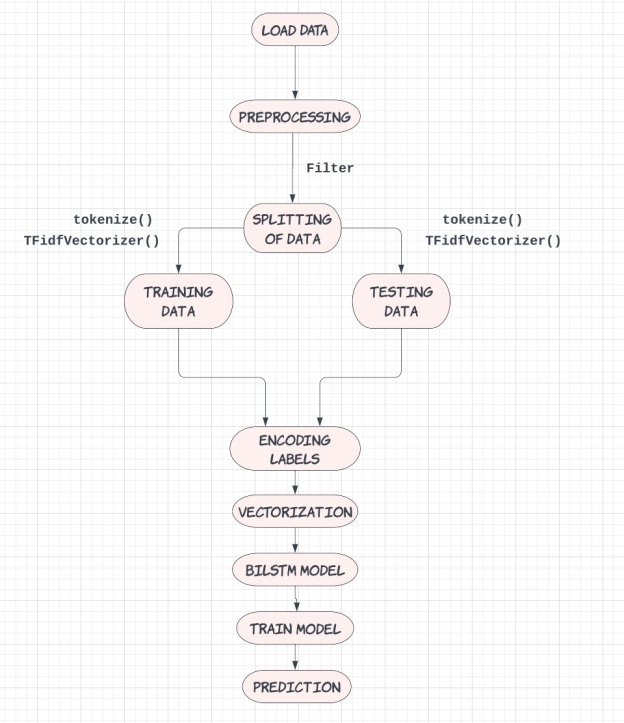
We want to find the best closest answer for the given question. Bi-LSTM models learn representations from data, eliminating the need for handcrafted features used in Learning to Rank approaches and allowing for more flexible training. It capture sequential information and semantic relationships,enabling better understanding of context and accurate answer Generation.

- The model is developed using the Python programming language and several popular libraries such as Keras, Tensorflow, scikit-learn, and nltk.
- The model takes a dataset of questions, answers, and labels as input, where the labels indicate whether the answer is a perfect match to the question or not. The dataset is preprocessed using various techniques such as text cleaning, stopword removal, and label encoding. Text cleaning involves converting the text

to lowercase, removing digits and special characters, and removing stopwords. Stopwords are common words in the English language that do not provide any valuable information for the model.

- The preprocessed data is then vectorized using the Term Frequency-Inverse Document Frequency (TF-IDF) method, which is a popular technique in natural language processing (NLP). The vectorization process converts the textual data into a numerical format that can be processed by the model. The model architecture is defined as a bidirectional Long Short-Term Memory (LSTM) neural network with an embedding layer and a dense layer with sigmoid activation. The LSTM is a type of neural network that is well suited for processing sequential data, such as text. The embedding layer is used to learn a vector representation of the input text, which is used as input to the LSTM. The sigmoid activation function is used in the output layer to predict the probability of the answer being a perfect match to the input question.
- The model is trained on the preprocessed data using binary cross-entropy loss and the Adam optimizer. Accuracy is used as the evaluation metric to monitor the performance of the model during training. An early stopping callback is used to prevent overfitting, where the model is stopped from training if the validation loss does not improve after a certain number of epochs.
- Finally, a prediction function is defined to predict the answer to a given input question. The prediction function preprocesses the input question and vectorizes it using the same methods as the training data. Then, it computes the cosine similarity between the input question and the preprocessed questions in the dataset. The answer with the highest cosine similarity score that has a label of 1 (i.e., a perfect match to the question) is returned as the output. If no perfect match is found, the function returns the three closest answers in terms of cosine similarity score. If no match is found, a default response is given to contact the provider by phone or email. Overall, this code demonstrates a powerful application of machine learning and NLP techniques for text classification, specifically for question-answering tasks.

6.2.3 Model 3 - BiLSTM + Category:



Since Model 2 took more computational time, we tried to integrate Model to predicting the category. The run time taken has drastically reduced from two minutes by previous Bi-directional LSTM to five seconds because we first predict categories and then run Bi-directional LSTM algorithm.

The main difference between the two models is the data that is being used. This model filters the data by category before processing and training the model, while the previous model uses the original data as is.

Other than that, the two models are very similar in terms of the preprocessing steps, vectorization, and the BiLSTM model used. They both clean the text data by removing numbers and special characters, stop words, and encode the labels. They also use a TfidfVectorizer for vectorization and a Bidirectional LSTM model for classification. Finally, they both use cosine similarity to find the closest answers to a given question and return the perfect answer if available, or the closest available answers if not.

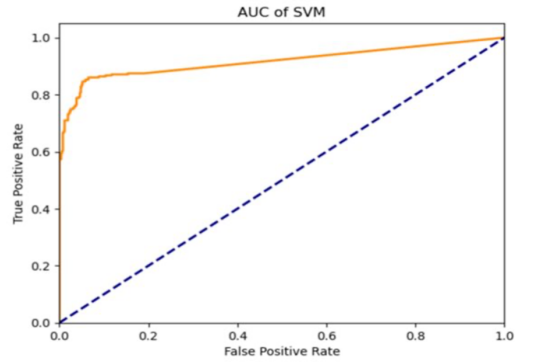
7 Analysis of Experiment results:

7.1 Model 1 - SVM

- Model with input 1:

	precision	recall	f1-score	support
0	0.89	0.96	0.93	521
1	0.91	0.79	0.85	279
accuracy			0.90	800
macro avg	0.90	0.87	0.89	800
weighted avg	0.90	0.90	0.90	800

AUC: 91.67% , PRC: 92.25%



According to the given categorization report, the model's overall test set accuracy was 90%. The accuracy of class 0 is 89%, which indicates that 89% of the instances expected to be class 0 were in fact class 0. In other words, 96% of the occurrences that were actually in class 0 were accurately predicted by the model, according to the recall for class 0.

The accuracy of class 1 is 91%, which means that, of all the cases predicted to be in class 1, 91% actually were. 79% of the cases that were really in class 1 were accurately predicted by the model, according to the recall for that class. The harmonic mean of precision and recall for class 0 is represented by the F1-score, which is 0.93 for class 0 and 0.85 for class 1. The average F1 score for both classes, or the macro average F1 score, is 0.89. The weighted average F1-score, which is the sum of the F1-scores for the two classes, is 0.90.

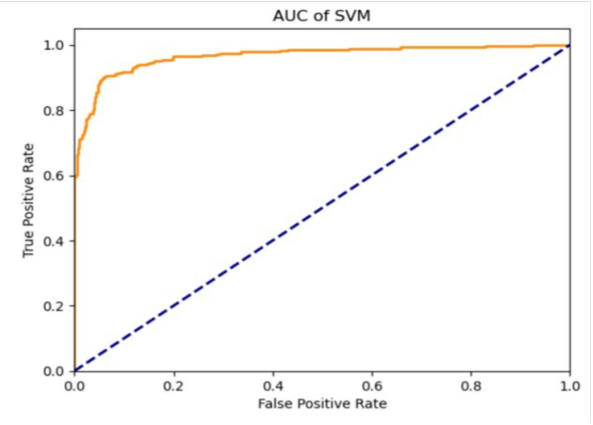
The AUC (Area Under the ROC Curve), which measures how well the model distinguishes between the two classes, is 91.67%. The PRC (Precision-Recall Curve) is 92.25%, making it a useful indicator for datasets with an unequal number of examples of each class. Overall, the model performs well, although more research may be needed to enhance its performance in the class 1.

- Model with input 2:
  - No remove stop-words:

```
tfidf_vector,clf=svm_model(X_train,Y_train,X_test,Y_test,\
                           remove_stopword=False,bigram = False)
```

	precision	recall	f1-score	support
0	0.92	0.95	0.94	521
1	0.90	0.85	0.88	279
accuracy			0.92	800
macro avg	0.91	0.90	0.91	800
weighted avg	0.92	0.92	0.92	800

AUC: 96.62% , PRC: 95.41%



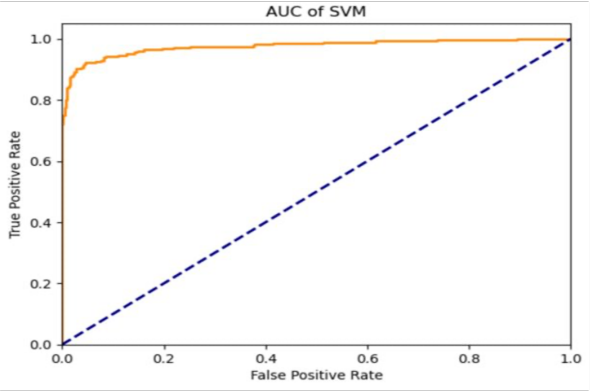
Based on the classification report, the model achieved an overall accuracy of 92% on the test data. In this case, the AUC score is 96.62%, which is a good indication that the model is effective at distinguishing between the two classes. The PRC score measures the trade-off between precision and recall, with a value of 1.0 indicating perfect precision and recall. In this case, the PRC score is 95.41%, which is also a good indication that the model is effective at identifying positive and negative sentiments.

- Remove stop-works:

```
tfidf_vector_1,clf_1=svm_model(X_train,Y_train,X_test,Y_test,\
                           remove_stopword=True,bigram = False)
```

	precision	recall	f1-score	support
0	0.93	0.98	0.96	521
1	0.97	0.87	0.92	279
accuracy			0.94	800
macro avg	0.95	0.93	0.94	800
weighted avg	0.95	0.94	0.94	800

AUC: 97.61% , PRC: 97.10%



The results show better performance in almost all metrics compared to the above case. In terms of AUC and PRC, this case also performs better, with an AUC of 97.61% and a PRC of 97.10%. Overall, removing stop-words indicates better performance in detecting class 1 (positive) instances and has higher overall model performance.

- Evaluate the performance of the retrieve document with the top-1, top-3, and top-5 answers for 2 cases.

```
score = test_retrive(final_dataset["Question Chat GPT"], top =1,\
            remove_stopword=False,bigram = False,change_para=True)
print("Accuracy for the top 1 is ", sum(score)/len(score)*100)
```

```
Accuracy for the top 1 is  41.0
```

```
score = test_retrive(final_dataset["Question Chat GPT"], top =3,\
            remove_stopword=False,bigram = False,change_para=True)
print("Accuracy for the top 3 is ", sum(score)/len(score)*100)
```

```
Accuracy for the top 3 is  63.0
```

```
score = test_retrive(final_dataset["Question Chat GPT"], top =5,\
            remove_stopword=False,bigram = False,change_para=True)
print("Accuracy for the top 5 is ", sum(score)/len(score)*100)
```

```
Accuracy for the top 5 is  75.0
```

Case 1: No removing stop-words

```
score = test_retrive(final_dataset["Question Chat GPT"], top =1,\
            remove_stopword=True,bigram = False,change_para=False)
print("Accuracy for the top 1 is ", sum(score)/len(score)*100)
```

```
Accuracy for the top 1 is  48.0
```

```
score = test_retrive(final_dataset["Question Chat GPT"], top =3,\
            remove_stopword=True,bigram = False,change_para=False)
print("Accuracy for the top 3 is ", sum(score)/len(score)*100)
```

```
Accuracy for the top 3 is  67.0
```

```
score = test_retrive(final_dataset["Question Chat GPT"], top =5,\
            remove_stopword=True,bigram = False,change_para=False)
print("Accuracy for the top 5 is ", sum(score)/len(score)*100)
```

```
Accuracy for the top 5 is  75.0
```

Case 2: Removing stop-words

In case 1, the accuracy for retrieving the correct document in the top 1 result is relatively low, at only 41%. However, the accuracy improves as we consider more results, with the top 3 and top 5 accuracies being 63% and 75% respectively.

In case 2, the accuracy for retrieving the correct document in the top 1 result is higher, at 48%. The accuracy also improves as we consider more results, with the top 3 and top 5 accuracies being 67% and 75% respectively.

Overall, case 2 performs better than case 1 in terms of retrieving the correct documents. However, the overall accuracy of retrieving the correct documents is still relatively low, indicating the need for improvement in the retrieval method. One of the reasons could be from predicting the category for the question. Although the model shows high accuracy at 93%, the dataset used to train this model seems biased. We augmented the dataset by adding, replacing, and deleting words from the original 330 questions, which did not change the general meaning and words of the questions. Hence, when we use the synonym questions from ChatGPT, it provides new words that the model has not been trained on, resulting in predicting the wrong category and retrieving the wrong answers.

7.2 Model 2 - BiLSTM:

The experiment involved two models: Category Detection and BiLSTM Question-Answering. The Category Detection model utilized Decision Tree, Random Forest, Logistic Regression, and XGBoost algorithms. Random Forest showed the highest performance, making it the chosen model. The BiLSTM Question-Answering model used TF-IDF vectorization and a Bidirectional LSTM neural network. The experiment achieved good results, accurately categorizing questions and providing relevant answers. However, some questions were misclassified due to similar keywords. Overall, the models performed well in classifying and answering questions, demonstrating their effectiveness in information retrieval. Future improvements could focus on addressing keyword similarity issues and introducing a “No category” or “New category” option for unmatched questions. The experiment highlights the potential of machine learning models in automating customer support and enhancing user experience.

8 Conclusion and future work:

In conclusion, we have scope for NLG FOR Chatbot. The results of this project indicate that automating question categorization and answering for university context has great potential. The SVM and Random Forest models demonstrated good performance in category detection, while the SVM and BiLSTM models were effective in predicting answers based on question similarity. However, there is still room for improvement, such as incorporating keyword importance and developing solutions for cases where no suitable category or answer is found. Overall, this project showcases the potential of machine learning models to enhance the efficiency and effectiveness of customer support in academic institutions.

Integration with chatbots: The models can be integrated with chatbots to provide quick and efficient support to customers. This could help reduce the workload on human customer support representatives and improve customer satisfaction. Multilingual support: The models can be trained on multilingual data to provide support in multiple languages. This would be particularly useful for companies operating in regions with diverse linguistic backgrounds. Real-time updates: The models can be updated in real-time as new questions and answers are added to the database. This would help ensure that the models remain accurate and up-to-date. Incorporating sentiment analysis: this can help improve the models by allowing them to identify and respond to customers’ emotional state, leading to a more personalized and empathetic customer support experience. This can be achieved by adjusting the ranking of answers based on the sentiment input from the customer.

9 Project task list:

Task	Assignee	Signature.
(1) Data collection	Han, Manvendra	
(2) Data preprocessing	Yashita	
(3) Model 1	Han	
(4) Model 2	Yashita, Manvendra	
(5) Topic Analysis	Yashita, Han	
(6) Hypothesis testing and interpretation	Manvendra	
(7) Poster creation	All	
(8) Research report writing	All	

10 References:

1. https://link.springer.com/chapter/10.1007/978-3-030-49186-4_31
2. https://www.researchgate.net/profile/MenalDahiya/publication/321864990_A_Tool_of_Conversation_Chatbot/links/5a360b02aca27/247eddea031/A-Tool-of-Conversation-Chatbot.pdf
3. https://arxiv.org/pdf/1612.01627.pdf
4. https://www.researchgate.net/profile/Darius-Zumstein/publication/322855718_Chatbots_An_Interactive_Technology_for_Personalized_Communication_Transactions_and_Services/links/5a72ecde458515512076b406/Chatbots-An-Interactive-Technology-for-Personalized-Communication-Transactions-and-Services.pdf
5. https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Chatbot%20for%20university%20related%20FAQs
6. https://ieeexplore.ieee.org/document/8126057
7. https://www.researchgate.net/publication/362546056_Chatbot_for_College_Website
8. B. R. Ranoliya, N. Raghuwanshi and S. Singh, "Chatbot for university related FAQs," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 2017, pp. 1525-1530, doi: 10.1109/ICACCI.2017.8126057.
9. https://mmuratarat.github.io/2019-10-12/probabilistic-output-of-svm