Ashwini Doke

Student ID: 20493897

A study on a chatbot allowing students to quickly schedule part-time jobs.

Table Of Content

1. Introduction

The use of chatbots has become increasingly prevalent in various industries, providing a convenient and efficient way for individuals to interact with businesses and organizations. In the education sector, a chatbot for shift booking system for students can provide a useful tool for managing work schedules and responsibilities. This report will provide an overview of the potential benefits and capabilities of a chatbot for shift booking system for students, as well as discuss the potential challenges and limitations of such a system.

While booking for part time shifts at an agency, we get emails or messages about shift availability, but most of the times we don't get the shifts due to a huge traffic of responses to the managers and we end up with no shifts to work. This way of booking the shifts is becoming very redundant for aspiring students and it inspired me to create a chatbot through which a student can book the shift at any location, time and place they want with ease and with fast response.

JobSeekers aims to be a part-time shift booking system for aspiring students who want to earn and learn as they go. The students will be able to check the available shifts, book any shift they want, cancel or review their prebooked shifts at an ease from wherever they want, without having to call the managers for checking the available shifts or even waiting for getting a confirmation of booked shifts. The managers will also be relived from a lot of redundant tasks of checking on availability of shifts, student's schedules, capability of students for the job and many more manual tasks

2. Current application and opportunities

The study "Designing a text-based AI scheduling assistant chatbot for a business environment" by HAU-BEN BENJAMIN SHIH [1] describes the development of a chatbot system for scheduling meetings and appointments in a business setting. The chatbot uses natural language processing, conversational user interface to understand requests from users and provide them with options for available times and locations for meetings. This paper also discusses about the history of chatbots, their evolution and different types of chatbot and their pros and cons.

Our chatbot system for booking part-time shifts is similar to the one described in Shih's [1] study in that it uses natural language processing to assist users in scheduling their work shifts. However, there are several key differences between the two systems. Firstly, our chatbot is designed for students, whereas the chatbot described in Shih's study is intended for use in a business environment. This means that our chatbot takes into account the unique challenges and needs of students, such as their busy school schedules and the need to balance work and study.

Another key difference between our chatbot and the one described in Shih's [1] study is the way in which the chatbots interact with users, in the study they have used a menu and text-based interaction. Our chatbot uses conversational AI technology to engage in more natural and human-like conversations with students, which makes it easier for them to use the chatbot and improves their overall experience. In contrast, the chatbot described in the study uses a conversational User Interface to interact with users.

Overall, our chatbot system for booking part-time shifts is similar to the one described in Shih's study [1] in its use of natural language processing but differs in its focus on students, and its ability to engage in natural and human-like conversations with users. These differences make our chatbot a valuable tool for students seeking to manage their work schedules.

Moreover, the 'Nazief & Adriani Stemming Algorithm With Cosine Similarity Method For Integrated Telegram Chatbots With Service' a study by Prismana , Prehanto , Dermawan , Herlingga and Wibawa [2] implements the Nazief & Adriani algorithm with the method cosine similarity using the PHP programming language and PostgreSQL database to create a website and bot. Website is used to enter FAQ whereas bot will provide responses or answers to incoming questions with the help of application services Telegram.

Unlike our chatbot which is implement using NLP's cosine similarity algorithm, is an independent conversational AI system, the study [2] has hosted the chatbot on Telegram.

3. Proposed System

This system keeps a SQL database with the information on the students, open shifts, and their booking. It responds to the student's inquiry using the cosine similarity measure and uses the "intents.json" file to match the intents. The algorithm selects the response that has the greatest degree of similarity and uses it to respond to the student user.

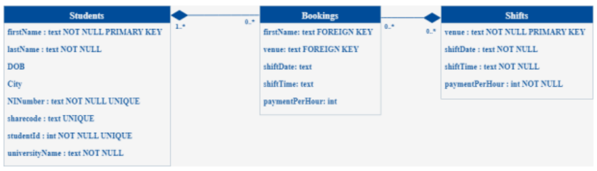
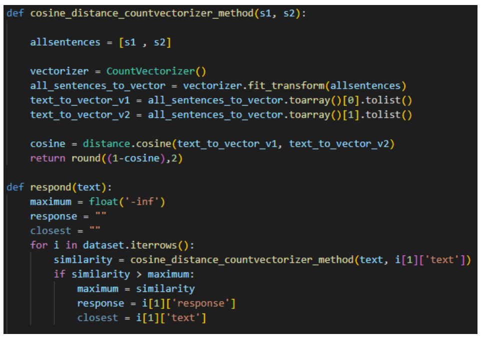
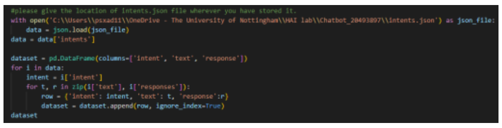

Students
firstName : text NOT NULL PRIMARY KEY
lastName : text NOT NULL
DOB
City
NINumber : text NOT NULL UNIQUE
sharecode : text UNIQUE
studentId : int NOT NULL UNIQUE
universityName : text NOT NULL
1..*
0..*
Bookings
firstName: text FOREIGN KEY
venue: text FOREIGN KEY
shiftDate: text
shiftTime: text
paymentPerHour: int
0..*
0..*
Shifts
venue : text NOT NULL PRIMARY KEY
shiftDate : text NOT NULL
shiftTime : text NOT NULL
paymentPerHour : int NOT NULL

Fig 3.1 Database structure



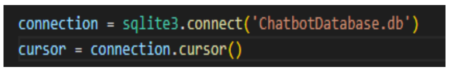
```
def cosine_distance_countvectorizer_method(s1, s2):

    allsentences = [s1 , s2]

    vectorizer = CountVectorizer()
    all_sentences_to_vector = vectorizer.fit_transform(allsentences)
    text_to_vector_v1 = all_sentences_to_vector.toarray()[0].tolist()
    text_to_vector_v2 = all_sentences_to_vector.toarray()[1].tolist()

    cosine = distance.cosine(text_to_vector_v1, text_to_vector_v2)
    return round((1-cosine),2)

def respond(text):
    maximum = float('-inf')
    response = ""
    closest = ""
    for i in dataset.iterrows():
        similarity = cosine_distance_countvectorizer_method(text, i[1]['text'])
        if similarity > maximum:
            maximum = similarity
            response = i[1]['response']
            closest = i[1]['text']
```

Fig 3.2 Cosine similarity



```
#please give the location of intents.json file wherever you have stored it.
with open('C:\\Users\\psxad11\\OneDrive - The University of Nottingham\\HAI lab\\Chatbot_20493897\\intents.json') as json_file:
    data = json.load(json_file)
data = data['intents']

dataset = pd.DataFrame(columns=['intent', 'text', 'response'])
for i in data:
    intent = i['intent']
    for t, r in zip(i['text'], i['responses']):
        row = {'intent': intent, 'text': t, 'response':r}
        dataset = dataset.append(row, ignore_index=True)
dataset
```

Fig 3.3 loading intent.son and creating dataset



```
connection = sqlite3.connect('ChatbotDatabase.db')
cursor = connection.cursor()
```

Fig 3.4 Establishing Database connection

This system uses Cosine similarity measure, which measures similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. It is often used in natural language processing (NLP) as a similarity measure between two pieces of text.

Some other algorithms that are commonly used in NLP include:

Jaccard similarity, which calculates the similarity between two sets by measuring the ratio of the intersection of the sets to the union of the sets.

Euclidean distance, which calculates the distance between two points in Euclidean space.

Levenshtein distance, which calculates the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one string into another.

One reason why cosine similarity is often preferred over these other algorithms in the context of NLP is that it is not sensitive to the overall magnitude of the vectors being compared. For example, if two pieces of text have the same words but one has the words repeated more times, the Euclidean distance between the two would be different from the distance between the same words with the same frequency, even though the semantic content of the text is the same. Cosine similarity, on the other hand, would produce the same similarity score for both cases, as it only considers the angle between the vectors, not their magnitude.

Cosine similarity gives excellent results for matching two documents, according to research by Gunawan (2018) [3] with the title "The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents."

Here's how the system work:

1. Registered students can review the list of open shifts, including their times and dates as well as the compensation per hour.

2. They can reserve any shift they like in accordance with their schedule of classes, homework, and exams from whatever location they choose.

3. They can also verify the shifts they have already reserved and cancel their shifts.

4. If a hazy response is found, the system will inform the user of its capabilities and how it works.

5. Furthermore, it will return greetings.

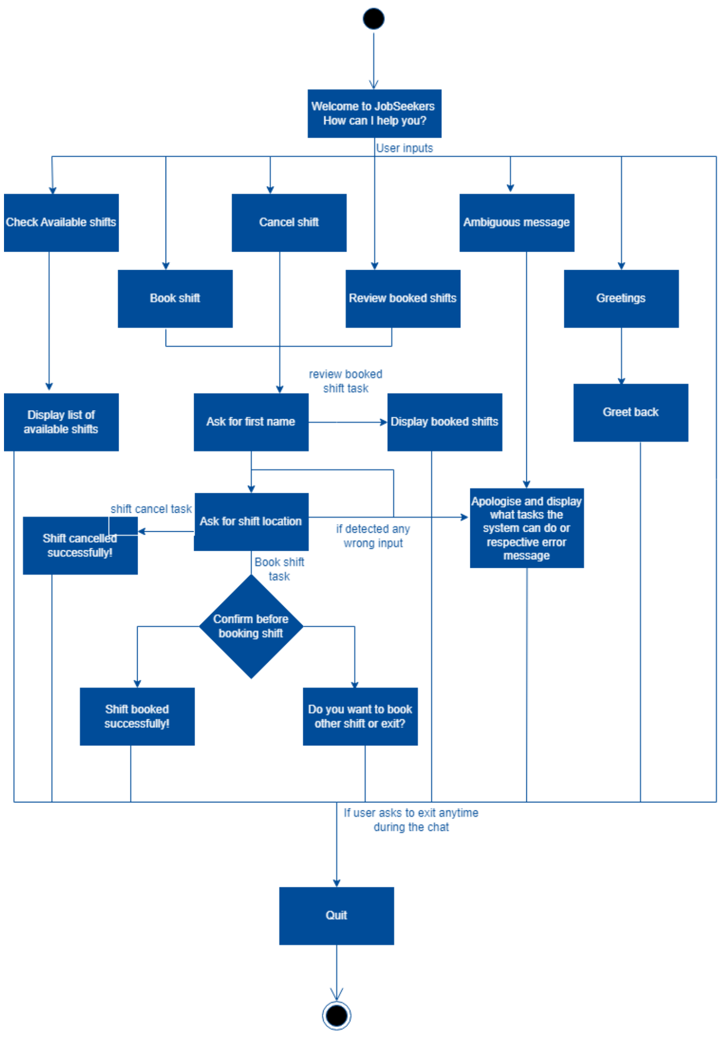
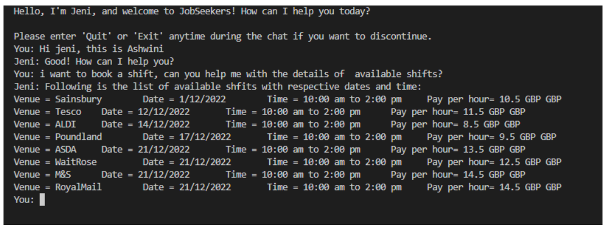

Welcome to JobSeekers How can I help you?
User inputs
Check Available shifts
Cancel shift
Ambiguous message
Book shift
Review booked shifts
Greetings
review booked shift task
Display list of available shifts
Ask for first name
Display booked shifts
Greet back
shift cancel task
Shift cancelled successfully!
Ask for shift location
if detected any wrong input
Apologise and display what tasks the system can do or respective error message
Book shift task
Confirm before booking shift
Shift booked successfully!
Do you want to book other shift or exit?
If user asks to exit anytime during the chat
Quit

Fig 3.5 Activity diagram depicting the chatbot system

4. Evaluation

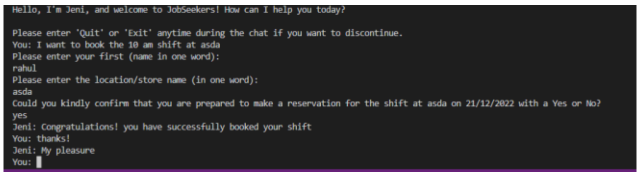
To evaluate the chatbot unit testing was carried out for each functionality.

a. Check available shifts



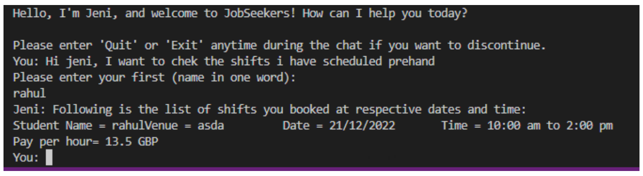
```
Hello, I'm Jeni, and welcome to JobSeekers! How can I help you today?

Please enter 'Quit' or 'Exit' anytime during the chat if you want to discontinue.
You: Hi jeni, this is Ashwini
Jeni: Good! How can I help you?
You: i want to book a shift, can you help me with the details of  available shifts?
Jeni: Following is the list of available shfits with respective dates and time:
Venue = Sainsbury        Date = 1/12/2022        Time = 10:00 am to 2:00 pm      Pay per hour= 10.5 GBP GBP
Venue = Tesco    Date = 12/12/2022       Time = 10:00 am to 2:00 pm      Pay per hour= 11.5 GBP GBP
Venue = ALDI     Date = 14/12/2022       Time = 10:00 am to 2:00 pm      Pay per hour= 8.5 GBP GBP
Venue = Poundland        Date = 17/12/2022       Time = 10:00 am to 2:00 pm      Pay per hour= 9.5 GBP GBP
Venue = ASDA     Date = 21/12/2022       Time = 10:00 am to 2:00 pm      Pay per hour= 13.5 GBP GBP
Venue = WaitRose         Date = 21/12/2022       Time = 10:00 am to 2:00 pm      Pay per hour= 12.5 GBP GBP
Venue = M&S      Date = 21/12/2022       Time = 10:00 am to 2:00 pm      Pay per hour= 14.5 GBP GBP
Venue = RoyalMail        Date = 21/12/2022       Time = 10:00 am to 2:00 pm      Pay per hour= 14.5 GBP GBP
You: 
```

b. Book a shift



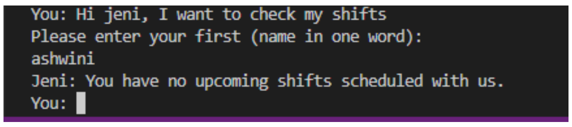
```
Hello, I'm Jeni, and welcome to JobSeekers! How can I help you today?

Please enter 'Quit' or 'Exit' anytime during the chat if you want to discontinue.
You: I want to book the 10 am shift at asda
Please enter your first (name in one word):
rahul
Please enter the location/store name (in one word):
asda
Could you kindly confirm that you are prepared to make a reservation for the shift at asda on 21/12/2022 with a Yes or No?
yes
Jeni: Congratulations! you have successfully booked your shift
You: thanks!
Jeni: My pleasure
You: 
```

c. Check my booked shifts



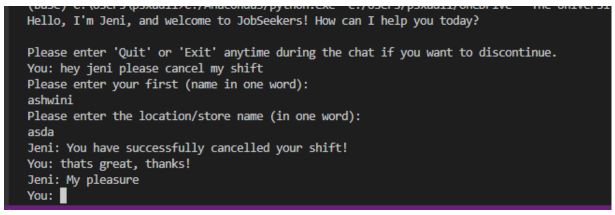
```
Hello, I'm Jeni, and welcome to JobSeekers! How can I help you today?

Please enter 'Quit' or 'Exit' anytime during the chat if you want to discontinue.
You: Hi jeni, I want to chek the shifts i have scheduled prehand
Please enter your first (name in one word):
rahul
Jeni: Following is the list of shifts you booked at respective dates and time:
Student Name = rahulVenue = asda         Date = 21/12/2022       Time = 10:00 am to 2:00 pm
Pay per hour= 13.5 GBP
You: 
```

d. Check my booked shifts, when there were none



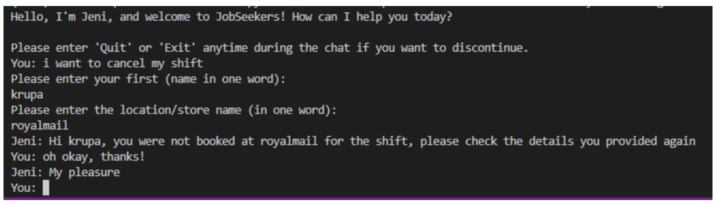
```
You: Hi jeni, I want to check my shifts
Please enter your first (name in one word):
ashwini
Jeni: You have no upcoming shifts scheduled with us.
You: 
```

e. Cancel shift



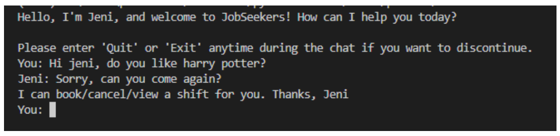
```
Hello, I'm Jeni, and welcome to JobSeekers! How can I help you today?

Please enter 'Quit' or 'Exit' anytime during the chat if you want to discontinue.
You: hey jeni please cancel my shift
Please enter your first (name in one word):
ashwini
Please enter the location/store name (in one word):
asda
Jeni: You have successfully cancelled your shift!
You: thats great, thanks!
Jeni: My pleasure
You: 
```

f. Cancel Shifts but there were no shifts booked



```
Hello, I'm Jeni, and welcome to JobSeekers! How can I help you today?

Please enter 'Quit' or 'Exit' anytime during the chat if you want to discontinue.
You: i want to cancel my shift
Please enter your first (name in one word):
krupa
Please enter the location/store name (in one word):
royalmail
Jeni: Hi krupa, you were not booked at royalmail for the shift, please check the details you provided again
You: oh okay, thanks!
Jeni: My pleasure
You: 
```

g. handle ambiguity



```
Hello, I'm Jeni, and welcome to JobSeekers! How can I help you today?

Please enter 'Quit' or 'Exit' anytime during the chat if you want to discontinue.
You: Hi jeni, do you like harry potter?
Jeni: Sorry, can you come again?
I can book/cancel/view a shift for you. Thanks, Jeni
You: 
```

All the negative and positive testcases were successfully tested. Overall, the evaluation of our part-time shift booking system chatbot should provide a comprehensive assessment of the chatbot's capabilities and performance and can help us to identify areas for improvement and optimization. This information can then be used to make necessary adjustments and enhancements to the chatbot to ensure that it continues to provide a valuable and effective tool for students.

A chatbot used to book part-time shifts can be a useful tool for students, as it allows them to easily schedule and manage their work hours around their busy school schedules. With a chatbot, students can quickly view available shifts, book the ones that work best for them, and receive updates about any schedule changes. This can save students time and effort, as they don't have to constantly check in with their employer to see if there are any open shifts or changes to their schedule. Additionally, a chatbot can provide students with easy access to information about their work schedule, such as the hours they have worked and the pay they have earned, making it easier for them to manage their finances. Overall, a chatbot can help students to balance their studies and work and make their lives a little bit easier.

However, it is important to consider the overall fairness and potential sources of bias in the chatbot. For example, the system may be more likely to offer certain shifts to certain students based on factors such as their location or previous work experience. This could lead to unequal access to opportunities for students, and it is important to ensure that the chatbot is designed in a way that does not discriminate against any particular group of students. Additionally, the chatbot's decision-making process should be transparent and explainable, so that students can understand why they are being offered certain shifts and have the opportunity to appeal any decisions they believe to be unfair. Overall, careful consideration of fairness and bias is essential in the development and deployment of a chatbot used to book part-time shifts.

In terms of improvements, we can extend the chatbot's functionality to include the ability to register new students. At the moment, the chatbot can only be used to book, cancel, view, and review shifts. Another potential future scope is the capability of providing shift notifications.

The ability for employers to add shifts to the database can also be provided as a new user type.

5. Conclusion

In conclusion, a chatbot for shift booking system for students can provide a convenient and efficient way for students to manage their work schedules. The chatbot can assist with tasks such as checking available shifts, booking and cancelling shifts. This can help students to better organize their time and responsibilities and can potentially improve the overall efficiency of the shift booking process. Overall, implementing a chatbot for shift booking system for students can provide numerous benefits and is worth considering.

6. References

[1] HAU-BEN BENJAMIN SHIH, ‘Designing a text-based AI scheduling assistant chatbot for a business environment.’, 2021, https://www.diva-portal.org/smash/get/diva2:1647530/FULLTEXT01.pdf.

[2] IGLPE Prismana , DR Prehanto , DA Dermawan , AC Herlingga , SC Wibawa. ‘Nazief & Adriani Stemming Algorithm With Cosine Similarity Method For Integrated Telegram Chatbots With Service’, 2021, https://www.researchgate.net/publication/351937909_Nazief_Adriani_Stemming_Algorithm_With_Cosine_Similarity_Method_For_Integrated_Telegram_Chatbots_With_Service.

[3] D Gunawan, C A Sembiring1 , M A Budiman, “The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents", 2018, https://iopscience.iop.org/article/10.1088/1742-6596/978/1/012120/pdf.