MACHINE LEARNING AND CHATTING SERVICES IN COLLEGE SEARCH: A STUDY ON STUDENTS' PREFERENCES AND SATISFACTION

Jiuyu Zhang[1], Sanskar Pokharel[2]

[1]La Jolla Country Day School, 9490 Genesee Ave, La Jolla, CA 92037
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

How does Machine Learning fit into the aspect of a highschooler searching for their dream college? It is widely known that being in high school is a stressful for most high school student, and creating something that can help them find the best college is important. Knowing that high school students like to participate in social activity, creating a chatting service that uses machine learning serves the purpose of helping them find their most suitable college without increasing the stress of their already stressful situation. To prove that this method is suitable for the situation, a survey asking 1000 high school students whether they would choose the chat option or traditional fill-out a form method, majority of them preferred the chat way. Based on the results supporting my thesis, I developed an application that will help high school students find their best college that consists of a chatting service that helps students achieve their goal with the help of Machine Learning.

KEYWORDS

College, Machine Learning, AI, High School

1. INTRODUCTION

Communicating with friends via a chat app is now incorporated into the daily life of a high school student, and finding the best colleges could be a challenge that is in the daily life of a high school student as well [1]. Creating an app that uses a feature that simulates a chatting environment can in fact help the student fight their challenge and can in fact make the college application easier. Using the popularity of using chat apps, it would also be easier for high school students to adapt to the new way of finding colleges. Since students already know how to use chatting software efficiently and easily, they would find it easy to ask the chatbot the questions that they want answers to [[2]. Using machine learning as the backbone of the chatbot is also beneficial to the user experience, as each message sent by the user will be analyzed by the bot and each time the user chats with the bot, the response will be more like a casual human being [3]. And based on government information, the bot automatically updates its college database with the up-to-date information, ensuring that the student gets the most up-to-date information about each college.

Some of the existing techniques and systems that have been proposed to help high school students find their dream college, which allows the user to fill out a long and extensive form to

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD- 2023
pp. 211-218, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130518

achieve the promised results. However, these proposals assume that students have the time to complete the form, which is rarely the case in practice. Other techniques, such as analyzing a quiz that they make students take. They failed in making this an accurate analysis because of the variation of the students' mood when they are taking the test, making their algorithm useless for accounting the error margins and often results in incorrect scoring. A second practical problem is that some users find it hard to understand the results returned by other software. Most college finder returns a wide array of colleges that they might be interested in, often needing the user to click into each to see details. This unnecessary step caused many to step away from the website and search for other easier services to use.

In college core, the chat feature stands out as the main tool to help the user achieve their goal. By using the power of Machine Learning, the chatbot understands the input of the user and respond with necessary questions to determine the information that the user wants. Our goal for the chat module in CollegeCore is to create a complete network of questions so that our users will have more information about what they need. This method is inspired by the automated support popup that some company (e.g. HP) have on their website, these pop ups are especially useful when we need help on something about the company [4]. Because of how helpful it is, having a similar option in the college counseling side could also be helpful to the user. Existing tools ask users to fill out a long form, and often having unnecessary results showing up after completing the form. CollegeCore's Chat feature, however, allows for a more fun and stress-free experience, and with the chat taking account of the user's datas, a more accurate and personalized result

By conducting a survey on US high schoolers of their opinion of using CollegeCore Chat versus normal form-based counseling, most survey takers claimed they prefer CollegeCore. This survey is conducted with a population size of the number of students from both a public and private high school, with the sample size of the students that are in 11th or 12th grade. Survey takers are asked to use college core to try to find the college he/she might be interested in, and after using college core, they are asked to use the normal form-based counseling method. After performing both tasks, they are asked on their preference, accuracy (whether both suits the student interest), and time (which method gives the result faster). The survey result is then divided into 11th grade pool and a 12th grade pool, with 11th graders in the 11th grade pool and 12th graders in the 12th grade pool. Both pools have similar result. Both showed preferences towards using CollegeCore Chat, and 82% of 11th graders have a better accuracy with CollegeCore, while 79% of 12th graders have a better accuracy with CollegeCore. All participants reported that CollegeCore chat returned the results faster. This survey reflected the advantage of usingCollegeCore. It should also be noted, that for both CollegeCore and form-style counseling services, the participants are all using fresh accounts to ensure fairness in the survey.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we faced during the experiment and during the design phase of the project; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related papers that is relevant to CollegeCore in Section 5. Finally, Section 6 gives the conclusion remarks and points out the project's future work.

2. CHALLENGES

To build the project, a few challenges have been identified as follows.

2.1. Appropriate and Accurate Data About the Colleges

Making a chatbot about college is useless without appropriate and accurate data about the colleges [5]. Without accurate college data, users will be given incorrect information about their dream college, and it would be harder for the user to apply and get accepted to it.To find most accurate data, I searched the government and education websites around the world, in attempt to find the most accurate dataset for the bot to use while it is chatting with the users. The final data used are combined data from multiple credible sources (sources from government websites or trustable research websites). These data were then converted to a JSON file and were loaded into our backend server, which will analyze the data given by the user and find the best suitable information in the dataset to return to the user [6].

2.2. Keeping the Data Up to Date

Keeping the data up to date is also important for the accuracy of datas. Datas of the thousands of colleges around the world change all the time, and it is important for them to be the most recent. But it is not good enough to update it with different patches with the application. It must be updated regularly. The backend of the applications, which handles all the machine learning and AI chatting operations, are used to instantly update the data. The updated data, just needs to be uploaded to the backend server, and the data update will be complete. The datas are frequently updated, and they are usually updated 2-3 times per month to ensure that the data is accurate and up to date. Some datas that are not updated for years are marked and upon next update they will be treated as important updates so it receives a much more comprehensive search.

2.3. Data Together

Analyzing said data together is also important for a good user experience. Without good organization on how to analyze the data, it would also be hard to providean accurate data. With python’s TensorFlow library, I was able to combine understanding the user’s request and the analysis to retrieve the best result possible to return to the user. With the result from the algorithm, CollegeCore can return the suitable and crucial information to the user, allowing them to plan their high school life better. With on-demand backend pushing, the features available to the chatting service will be fine-tuned and updated frequently, allowing more features at a lower device storage usage rate. The communication between the frontend and backend are also monitored and will be used to train the machine learning furthermore to ensure a better and more accurate understanding of the user’s intentions.

3. SOLUTION

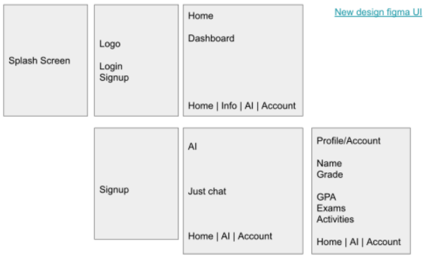
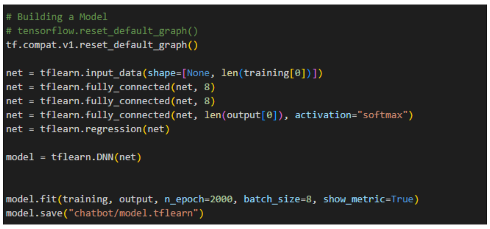


Figure 1. Overview of the solution

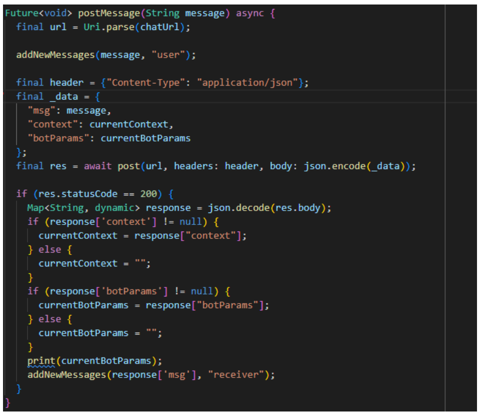
CollegeCore is a College Counseling app that aims to help the user, usually a middle or high schooler, to find their dream college and help them get through high school. As Figure 1 shows, users are first greeted with a Splash Screen which explains to them how the app works, then the user is led to login and sign-up page, where they can create their personalized CollegeCore account [7]. The user will then have access to our chatting system, where the user can then chat with our bot about anything, and the bot will try to answer as accurately as it can. The user could also edit their account information, such as their email, grade, and gpa, as the chat bot also consider the grade and gpa as a factor in producing the accurate answers. The signature feature, the Chat, consists of the following steps. The user first enters their message into the chat box, and press send. The message then gets embedded into a JSON object that contains other necessary information and sends it to the backend server. The backend then uses machine learning to convert the string of message to a specific category of answers, then using that category, the backend server analyzes it and then replies with the necessary information. If the backend needs extra information, it would ask the user about the question, and then process the returned information.



```
# Building a Model
# tensorflow.reset_default_graph()
tf.compat.v1.reset_default_graph()

net = tflearn.input_data(shape=[None, len(training[0])])
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, len(output[0]), activation="softmax")
net = tflearn.regression(net)

model = tflearn.DNN(net)


model.fit(training, output, n_epoch=2000, batch_size=8, show_metric=True)
model.save("chatbot/model.tflearn")
```

Figure 2. Building the Machine Learning Model

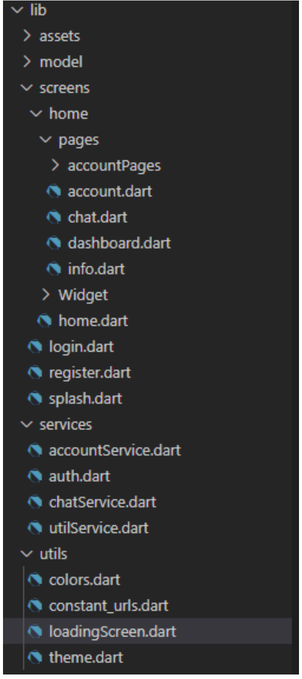
Figure 2 shows the necessary code to build a machine learning model with TensorFlow [8]. The model is trained 2000 times in this example, but the number will be changed based on how much data each training set contains. The more data the training set have, the less time it will be trained. However, if the trained data have more important and harder to train data, the times will be increased to make sure that the data is as accurate as possible.



```
Future<void> postMessage(String message) async {
  final url = Uri.parse(chatUrl);

  addNewMessages(message, "user");

  final header = {"Content-Type": "application/json"};
  final _data = {
    "msg": message,
    "context": currentContext,
    "botParams": currentBotParams
  };
  final res = await post(url, headers: header, body: json.encode(_data));

  if (res.statusCode == 200) {
    Map<String, dynamic> response = json.decode(res.body);
    if (response['context'] != null) {
      currentContext = response["context"];
    } else {
      currentContext = "";
    }
    if (response['botParams'] != null) {
      currentBotParams = response["botParams"];
    } else {
      currentBotParams = "";
    }
    print(currentBotParams);
    addNewMessages(response['msg'], "receiver");
  }
}
```

Figure 3. Communications to Backend

Figure 3 is a picture of how the chat messages gets sent to the backend for analysis. As the _data variable shows, each passage contains 3 different options. Msg is the message sent by the user, context is used to identify the category of the question/answer the user is sending, and bot params stores all the previous responses, if the category requires multiple responses. This structure allows for a better communication between the backend and the frontend, and it made sure that the correct info is obtained and sent to the user. The communications are established using HTTP Rest methods, over https protocol, ensuring secured communications and enhanced communications between the user and the server would be secure. The ensuring the security of the communications, it can be ensured that the personal datas, such as the answers to questions, would be encrypted and secured, so no people would illegally obtain the user's datas.



```
∨ lib
  > assets
  > model
  ∨ screens
    ∨ home
      ∨ pages
        > accountPages
        account.dart
        chat.dart
        dashboard.dart
        info.dart
      > Widget
      home.dart
    login.dart
    register.dart
    splash.dart
  ∨ services
    accountService.dart
    auth.dart
    chatService.dart
    utilService.dart
  ∨ utils
    colors.dart
    constant_urls.dart
    loadingScreen.dart
    theme.dart
```

Figure 4. File Structure

Figure 4 shows the file structure used to create the frontend of the CollegeCore App. There are 4 pages in total at the time of writing (dashboard, chat, info, and account), and each time a page is opened, the current page they are on is updated to the new page instead of creating a new page, therefore ensuring a more optimized app. Also, repeated usage of dart/flutter codes are stored for easier usage and better optimization, making it easier for us to develop and update the app. We use Firebase as our database, as our effort to try and make accounts and user datas more secure [9]. User's data are very secure under firebase becauseno one, including the developers, except for the user, will be able to see the password.

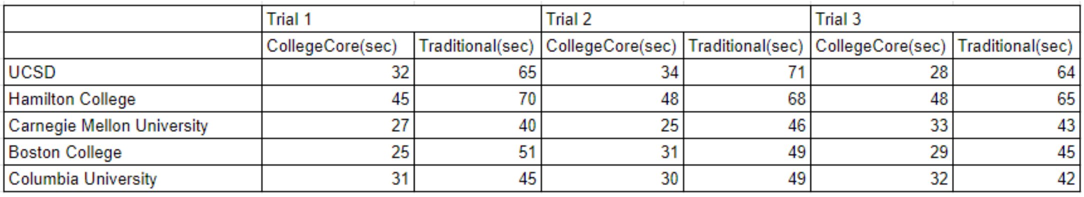
An app cannot look good without proper colors, and that's why CollegeCore have defined all the necessary colors in a single file, ensuring that they could be changed swiftly.If our users reported that they do not like the theme, or for general updates. Our users can provide feedback to us for anything and if they are valid, they will be fixed as soon as possible.

4. EXPERIMENT

4.1. Experiment 1

To verify that people prefer chat-style college planning app instead of the traditional form-based college planning app, we have set a random college in the US, and attempted to use a traditional form-based College planning app and CollegeCore to test the time needed and the number of questions needed for a user to get the result to that college. Each college will have 3 trials, with each trial containing the times needed to get that college for both methods. There will be 5 colleges in total.

Table 1. Results of experiment 1



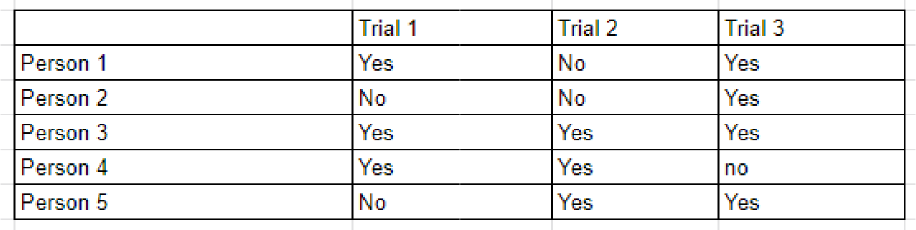
	Trial 1		Trial 2		Trial 3	
	CollegeCore(sec)	Traditional(sec)	CollegeCore(sec)	Traditional(sec)	CollegeCore(sec)	Traditional(sec)
UCSD	32	65	34	71	28	64
Hamilton College	45	70	48	68	48	65
Carnegie Mellon University	27	40	25	46	33	43
Boston College	25	51	31	49	29	45
Columbia University	31	45	30	49	32	42

The result of this experiment was not surprising, based on the premises of CollegeCore to simplify the process. The CollegeCore method is usually quicker than the Traditional method of trying to find a specific college, as seen in Table 1. For each trial and for each university, CollegeCore is usually around 20 seconds faster than the traditional methods. This, could be because of there are less repetitive questions being asked every time it is ran, because essential information, such as gpa, can be retrieved in the user profile. Eliminating a whole question because the information can be retrieved at any time, without having to ask. This removes at least 10 seconds from the total time, causing an easier and faster experience.

4.2. Experiment 2

Another way to make sure that CollegeCore is accurate is to see if the result match a person's dream college that they had before they used CollegeCore. This proves that collegecore is a better solution to college planning because it can returnan accurate result, even without all the necessary questions that traditional college planning apps have. There are 5 participants in this experiment, and each participant are asked to use college core 3 times without using the same answers for each trial. They are then asked on whether the app returned the correct college.

Table 2. Result of experiment 2



	Trial 1	Trial 2	Trial 3
Person 1	Yes	No	Yes
Person 2	No	No	Yes
Person 3	Yes	Yes	Yes
Person 4	Yes	Yes	no
Person 5	No	Yes	Yes

Of all the 15 responses of 5 randomly selected participant, ⅔ of the responses reported showed that they did indeed receive the right college. Although this ratio does not look like a bug percentage, considering the four thousand schools in the US, it would be hard to eliminate down to just 1 school for every trial. The negative options in the trials can also be explained because participants are asked to use different responses on each trial, showing how some of the different inputs can also led to the program returning a different school. Other factors that could result in a different school returned could be because of unrecognizable inputs the user might reply, or if the user have not fill out their profile data correctly.

These results showed how CollegeCore's chat feature could get its users results faster and more accurate than the normal services. In the first experiment, for 5 random schools from across the us, ALL can be obtained faster than the traditional methods, proving that for most universities, CollegeCore can save the user time in looking up the college they wanted, saving more time for the student to work on their school work. From the second experiment, it showed the accuracy of CollegeCore. As ⅔ of the trials showed a positive result, it can be said that the result provided given the shorter time it took to return the college, is accurate. As a result of the experiments, it can be said that the dream college of a user could be accurately returned within a shorter amount of time than the traditional, formed-based methods.

5. RELATED WORK

Using several types of machine learnings could help students in achieving career placements [10]. The author of this source went into a dive into the relation between Machine Learning (ML) and Explainable AI(XAI) and talked about how a model should be trained. However, my paper talked about the method and the best solution to optimize and solve a problem that the existing industry has. For example, Guleria and Sood talked about the Pre-modeling, Explainable modelling, and post-modelling stage of the Machine Learning model, while I focused more on a broader scale, talking about the general solution and the results of the tested solution.

This article, by Yang and Talha, focuses on how Big Data technology and Artificial Intelligence are affecting college counseling firms [11]. Yang and Talha’s paper shares similarities with this paper as they both studies the aspect of using machine learning in the field of college counseling, but Yang and Talha’s paper also talked into the Big Data technology, stepping beyond what this paper studied and explained. The consideration of big data might even make the college counseling apps even moreaccurate based on how many datas can be taken from the world and fitting it to predict the right answer, making it even a step over how CollegeCore handles it.

This paper is like my project as it discusses the client-server communications [12]. It showed structures showing how a typical client-server model and how this model benefits websites and applications, while also discussing the possible issues that the model has. This is exactly the structure that CollegeCore uses for front-backend communications. CollegeCore uses a server from Digital Ocean, a virtual private server(vps) provider that is trustworthy and uses a secure and private URL, with authentication checks built-in to ensure secure data transfer. This paper showed a clear view on how front to backend communications should be handled and how to make it secure and have optimized performance.

6. CONCLUSIONS

Overall, CollegeCore is an app designed to help high school students to find their dream college through Machine Learning and AI chat bot. The chat bot, will understands the dialogs between the bot and the user, and the backend will produce the accurate result that the user has requested. The Chat service is effective, as the method is tested in experiments, where the result showed that the chatbot can respond accurately to the user’s questions. The time-to-result experiment, which tested the time needed for college core to find the college and compare it to the times needed for traditional style college planner methods to find the college. The result is not surprising—CollegeCore were able to beat the traditional ones by 10-40 seconds. The second experiment conducted, the result-accuracy experiment, which asked participants to use collegecore and see if they receive the college the same as their dream college—two thirds of the trials turn out to be accurate. Considering there are as much as 4000 colleges in the US, achieving that rate is quite impressive. Analyzing the results of the two experiments, it can be concluded that if the user answers the questions the bot asks truthfully, the bot will return a close result to their dream college. With the chatting method, however, users are feeling less stress, as the method of the chat is simulating the users chatting with a friend, coming as a less intense way of asking questions.

The possible limitations of CollegeCore AI Chat include understanding of spelling errors in messages. If the messages sent have considerable number of misspelled words, the model might not be able to categorize the request, and therefore having the mark the message as indecisive. This could only be improved by providing a larger data and by training the model multiple times. Other limitations might include the data structure. Although the data is updated frequently, there are occasionally errors in college data's JSON format.

The limitations could be solved in the future by introducing newer technologies as they come up. With the rise of Big Data technology, for example, the accuracy of the machine learning model could be improved, and most importantly, the data for colleges could also be improved. Another way is to make user editions available. If a user spots an inaccurate detail, they could edit it, where they could be approved by developers and be updated if the detail is verified. This ensures the maximum accuracy of college datas.

REFERENCES

[1] Botha, Johnny, W. C. Vant, and Louise Leenen. "A comparison of chat applications in terms of security and privacy." Proc. 18th Eur. Conf. Cyber Warfare Secur.. 2019.

[2] Adamopoulou, Eleni, and Lefteris Moussiades. "An overview of chatbot technology." Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16. Springer International Publishing, 2020.

[3] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR). [Internet] 9 (2020): 381-386.

[4] Adair, Bill, et al. "Automated pop-up fact-checking: Challenges & progress." Proceedings of the Computation+ Journalism Symposium. 2019.

[5] Wilcox, Bruce, and Sue Wilcox. "Making it real: Loebner-winning chatbot design." ARBOR Ciencia, Pensamiento y Cultura 189.764 (2013): 1-13.

[6] Bangare, S. L., et al. "Using Node. Js to build high speed and scalable backend database server." International Journal of Research in Advent Technology 4 (2016): 19.

[7] Yilmaz, Yagiz, et al. "Investigating the impact of ransomware splash screens." Journal of Information Security and Applications 61 (2021): 102934.

[8] Abadi, Martín. "TensorFlow: learning functions at scale." Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming. 2016.

[9] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.

[10] Guleria, Pratiyush, and Manu Sood. "Explainable AI and machine learning: performance evaluation and explainability of classifiers on educational data mining inspired career counseling." Education and Information Technologies (2022): 1-36.

[11] Yang, Zhen, and Muhammad Talha. "A coordinated and optimized mechanism of artificial intelligence for student management by college counselors based on big data." Computational and Mathematical Methods in Medicine 2021 (2021).

[12] Oluwatosin, Haroon Shakirat. "Client-server model." IOSR Journal of Computer Engineering 16.1 (2014): 67-71.