CS/ECE 8803: Final Project. Design and implementation of a Georgia Tech MSECE curriculum advisor based on a LLM

Brieuc POPPER, Francisco Nicolás PÉREZ FERNÁNDEZ
bpopper3@gatech.edu, fn.perez@gatech.edu
Georgia Institute of Technology

***Abstract*—In this project, we compare different ways of creating an automated chatbot that acts as a curriculum advisor for Georgia Tech's graduate school of ECE. We came up with 5 different approaches: Using MS. Copilot (which uses RAG) with a custom prompt, and approaches based on the open-source Llama 7b LLM. We experiment with 4 different combinations of qLoRA finetuning and context feeding to this LLM. We conclude that it is difficult to teach facts to models through finetuning but that light finetuning can significantly improve the model's performance (+50% score with our custom metric). We also find that context is a convenient way of feeding unlabeled data to the model for it to quickly learn facts (a sort of zero-shot learning). However using context this way isn't ideal because of the huge memory requirements, longer inference times, and reliability issues. These problems justify the need for alternative techniques like Retrieval Augmented Generation (RAG) or finetuning on large, diverse datasets of good task-specific labelled data. Finally, we discuss the shortcomings that remain for each resulting models. Taking a step back, we find implausible that with current techniques an chatbot could be created that could be fully trusted on nontrivial curriculum advising tasks, especially with problems like LLM hallucination.**

***Index Terms*—Curricular advising task, LLM Fine-tuning, Labeled data generation, Context learning, Llama LLM applications, Georgia Tech utilities, chatbot, RAG, qLoRA**

I. INTRODUCTION

A wide repertoire of techniques has been discussed during this course when implementing specific NLP tasks: from the methods based on n-grams statistics, to the training of whole neural networks for inference of the outputs of the system. The recent literature studied in these last months, however, has showcased the benefits of both fine-tuning a large pre-trained language model (LLM) to a specific task, such as in BERT [1], as well as pre-loading the context before the user prompt to better align to the specific task and user expectations about the answer [2].

These methods allow us to save computing resources and labeled data quantity when implementing a specific task by utilizing a pre-trained weight set, with the weights optimized for language modeling through a set of general tasks, such as GPT-3 [3] or Llama 2 [4]. Extending this idea, different fine-tuned versions of the pre-trained weights on different tasks are published by the companies in the ML-NLP industry, so that these models can be further tuned with even less data and resources to satisfy even more specific tasks. This is the case for Llama 2 Chat [4], and this is the reason why this particular model is chosen as a starting point to implement the target task: to build a curriculum advisor for the Georgia Institute of Technology.

A. The curricular advising task

Different NLP tasks have been studied in this course. From the translation problems that gave rise to the encoder-decoder [5] and transformer [6] architectures, to the classification, summarization, and question-answering (QA) tasks that were strongly enhanced by the development of the first [1].

For this case, the curricular advising task (in the context of a particular program, the MSECE, from a particular university, Georgia Tech), can be defined as a composite task, requiring accurate answering to both open and closed questions, as well as generating curricula tailored to the student's request, based on a set of closed official sources.

The next section presents four approaches to the problem. These solutions are implemented on the Georgia Tech PACE ICE computer cluster and evaluated in section III, providing a comparison among them and with the Microsoft Copilot (formerly Bing Chat) results from CS/ECE 8803: Project #1 [7] [8].

II. METHODOLOGY

In this section, the four approaches to the curriculum advising task are presented in the same order as they were conceived, showcasing the motivation and rationale behind each solution.

A. First approach - Using Llama Chat without fine-tuning

The first approach, which is used mainly as a benchmark to compare vs. further refined models, is simply using Llama-7B-chat-HF directly while preceding the user prompt with a very simple context, only to prepare the model for answering questions in the context of Georgia Tech.

The instructions were a few sentences outlining that the model should act as a curriculum advisor. However, contrary to other approaches, no specific facts about GT's ECE grad school were provided (such facts could be a list of the courses available, their associated technical area of interest, how the thesis or non-thesis tracks work, the different rules for graduating, etc). This approach is depicted in Figure 1.

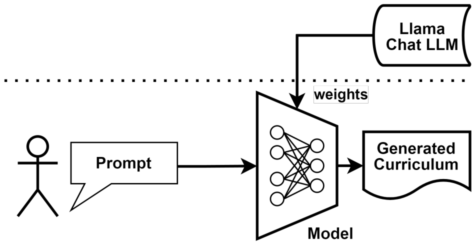

Fig. 1. First approach: Using the llama-2-7b-chat-hf.

B. Second approach - Fine-tuning with Georgia Tech official Course catalog

For this approach, **fine-tuning** was used. The following will go over a few theoretical and practical considerations for fine-tuning in the context of this project. The base model used is the Llama-2-7b-chat-hf. This model is already tuned to behave as an assistant and it is possible to have a conversation on a wide variety of topics. In the "evaluation" section later in this document, it is possible to compare this base model's performance to other more refined models built upon the base model discussed in the different approaches.

Recall that *fine-tuning* is something performed on a pre-trained model that is capable of a broad variety of tasks when the goal is to improve the performance of the model in a more restricted task. *Fine-tuning* specializes the model in the more restricted task with the hope of improving its performance, so fine-tuning is essentially just more training.

For most of the approaches that include fine-tuning, this model is trained with **supervised** fine-tuning with labeled data. Here the labeled data is an example prompt associated with its label: the expected text output in response to this prompt.

For this project, the fine-tuning technique used was LoRa[9]. This technique freezes the pre-trained model weights and injects trainable rank decomposition matrices into layers of the Transformers, which reduces the number of trainable parameters a lot and limits the GPU memory requirements, which was essential for our project given our computing resources. To further reduce the GPU memory requirements, we resorted to qLoRA which is based on LoRA with additional innovations, and it is extremely capable as it reduces memory usage enough to finetune a 65B parameter model on a single 48GB GPU while preserving full 16-bit finetuning task performance [10].

Our goal was to make the best possible "curriculum advisor chatbot" for Georgia Tech's ECE graduate students, and so we wanted to achieve some of these:

- Teach the model about ECE graduate school at Georgia Tech
- Ensure the model tries to be helpful, empathetic
- Ensure the model doesn't talk about anything apart from ECE curriculum

One of the interesting points explored in this project was trying to "teach knowledge" to our model with qLoRA [10] fine-tuning. Technically picking up new knowledge (as in for instance learning the course catalog for ECE grad school) is possible as fine-tuning is just a particular way of talking about more training, and the weights that are trained as part of the fine-tuning process can retain some information.

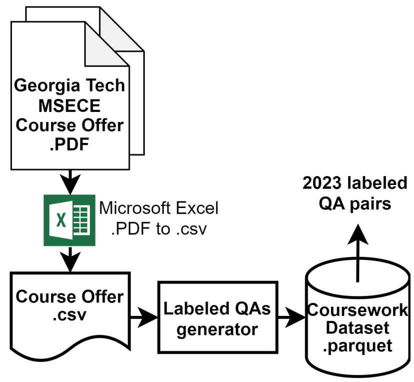

Fig. 2. Generation of labeled data from Georgia Tech official sources.

For this specific model, 2374 prompts and their associated label were created from a .csv file with the official Georgia Tech ECE course catalog. The questions were generated automatically from the CSV with a very simple mapping ("Labeled QAs generator in the Figure 2"): for example "What is course ECE XXXX" would be associated with "course ECE XXXX is *insert course name*". Here are some actual questions from the dataset of 2374 prompts:

- Can I register in ECE 6562: Autonomous Control of Robotic Systems in Spring 2024? No, ECE 6562: Autonomous Control of Robotic Systems can not be taken in Spring 2024
- How many credit hours is ECE 6456: Solar Cells? ECE 6456: Solar Cells is 3.0 student credit hours (SCH)
- and many others...

The experiment was to determine whether the model could learn the information contained in these questions. However, the dataset was not diverse at all (a lot of very similar questions, with the same phrasing) and it was very difficult to pick information up without overfitting and greatly losing most conversational skills.

To illustrate the potential negative effects, One of the earlier runs we did was completely unusable after the finetuning as it returned random nonsensical long lists of repetitive characters for no reason at all. Steps taken to fight against overfitting were increasing the dropout rate, and limiting the LoRA rank, but ultimately the dataset was not of sufficient quality for any improvements to the model's behavior.

Unfortunately splitting the data in a validation set and tracking the loss on training vs. validation data was not achieved, as this could have greatly helped understand the overfitting. However fig. 4 shows the loss during the training on a single epoch of all Q/A pairs. Again, this is probably overfitting a lot to the training data, but it is very difficult to generate a large dataset of quality supervised Q/A pairs, and it was still interesting to see the model's behavior on this. The

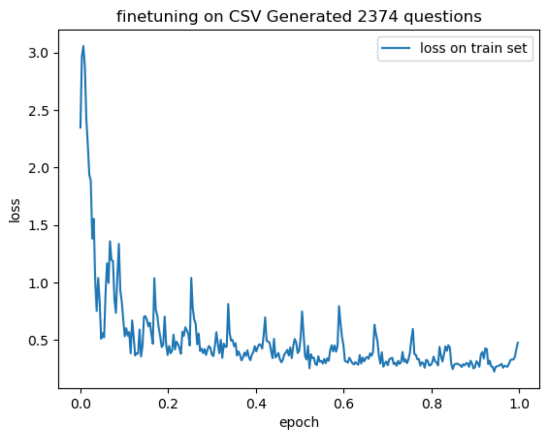

Fig. 3.

Fig. 4. Loss function for the qLora fine-tuning of Llama 2 Chat with 2374 labeled questions generated from Georgia Tech's course offer.

main parameters adjusted to prevent too much overfitting (as previous experiments showed that after overfitting the model could always answer huge nonsensical lists for no reason) were reducing the LoRA rank to 32 (this still weights more than 10 MB for the weights even though the training data fits in less than a MB which demonstrates potential for overfitting) and increasing the dropout rate. Interesting further experiments could have been with a very very low LoRA rank, even though it would have been hard for the model to "learn new knowledge" in that case, and with this dataset the finetuning would have probably ended up doing no good anyway. The training time for this fine-tuning was of about 15/20 minutes on a single GPU such as a Quadro RTX 6000 with 24GB of memory, for a single epoch (which was the whole training). It would probably be interesting to track carefully the model's loss, and overfitting, and tune the hyperparameters in more depth, however, the priority would first be to build a better dataset in terms of quality, diversity, and size. This is explored in further sections.

C. Third approach - Using only context

For this approach and the next, most of the 4096 available tokens supported by Llama will be used to feed it raw information about ECE courses, and the different rules that a curriculum advisor would need to have in mind. Having observed the difficulties for the model in the previous approach to remember the course information (i.e., combinations of course codes, course names, semesters where they are offered, and credit hours), this approach constitutes an experiment to explore the limits of compressing information in the context.

No fine-tuning is done here, it is just the Llama 7B HF chat model with a large context [1] provided to it initially. As mentioned, this context contains a synthesis of the registration requirements from the ECE Graduate Handbook, as well as all the graduate courses capable to fit from the Course Offer.

[1] Large Context

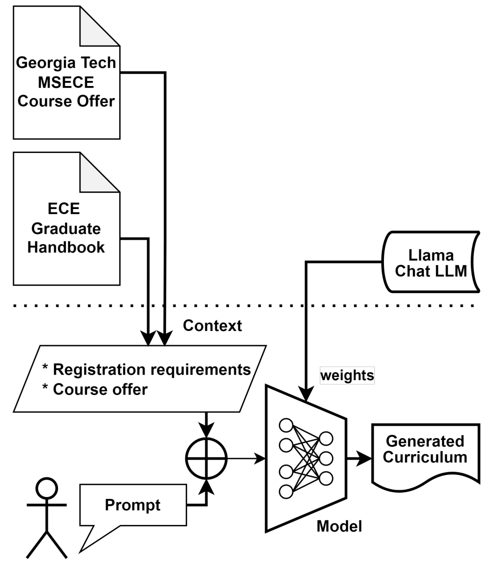

Fig. 5. Third approach: Loading a synthesized version of Georgia Tech official documentation (course offer plus registration rules) in the context.

D. Fourth approach, combining fine-tuning and context loading

As mentioned in the previous approach, the context is now used to its maximum.

Recall the detailed discussion on finetuning from the second approach. The main issue was the quality of the training dataset for supervised fine-tuning. In this approach which uses a whole new supervised dataset for fine-tuning, the goal is to create a "quality" dataset of about 100 questions. The generation of this dataset, as the Figure 6 showcases, is automatized by inputting a context (see annex C) to a prompt to an instance of Microsoft Copilot, until it generates reasonable quality outputs (based on the attached MSECE Course offer and Graduate Handbook). The results are manually reviewed (thus the low volume of answers), to improve quality, and then feeded with the original questions into a labeler script similar to the one used in the second case.

The goal of fine-tuning is not to "brute-force teach" new knowledge to the model but rather to orient its behavior to be closer to what is expected of a curriculum advisor. Some of the Q/A pairs teach the model how to format its answers for questions like "create a 3-semester plan to graduate from an ECE master, including which courses to take each semester.". Other Q/A pairs are taken from the publicly available FAQ from Georgia Tech's website and therefore are exactly the tone expected from a curriculum advisor.

The focus is really on the quality and diversity of the prompts, and the finetuning isn't designed to just learn a lot of hard facts, but rather to "orient" the model to being a better curriculum advisor, i.e., to align the model to the curriculum advising task.

Multiple combinations of LoRA ranks and dropout rates were chosen while monitoring the learning curves and the final

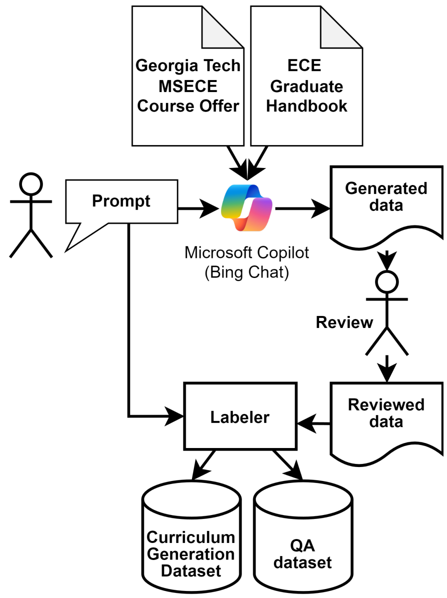

Fig. 6. Labeled dataset generation for both components of the course advising task: QA and curriculum generation

results, and finally good results were observed for 2 epochs, a dropout rate of .2 and a LoRA rank of 32.

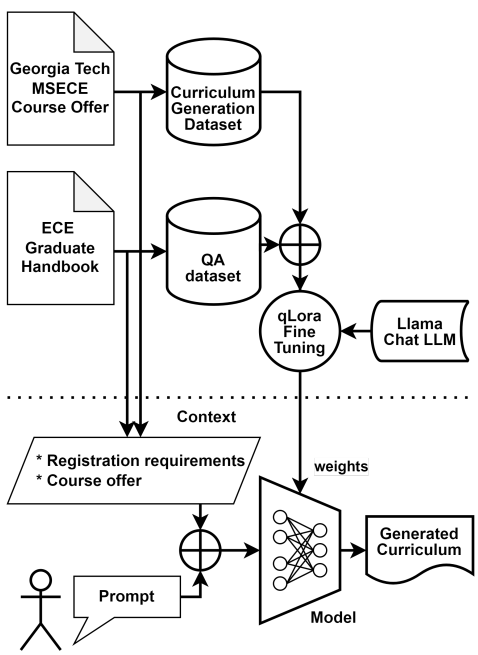

Fig. 7. Fourth approach, combining fine-tuning and context loading

III. Implementation Details

Every experiment from this project was created in Python with Huggingface's Transformers library with pytorch.

Different experiments were done with 1 to 4 GPUs from Georgia Tech's PACE ICE cluster. Learning how to properly use such a cluster and being able to run Python scripts requiring CUDA and powerful GPUs isn't easy, and it's interesting to note that a non-negligible amount of time in such projects is allocated to practical coding problems. However in the end we were able to run an interactive chat with a fine-tuned curriculum advisor smoothly, and we learned a lot in the process of getting there. The Llama 7b model can have a maximum of 4096 tokens of context, at which point it occupies about 32GB of video memory at inference time. To generate an answer, depending on how much context is used and the length of the answer, the model takes from 2 to 30 seconds. Loading the model also takes about 25 seconds.

For fine-tuning, the "Second approach" section discusses some of the specifics of practical implementation.

IV. Results

In this section, three evaluation metrics are proposed for comparing the four proposed solutions as well as the outputs from Microsoft Copilot (formerly Bing Chat, previously studied in CS/ECE 8803: Project #1 [7] [8]).

A. Evaluation metrics and methodology

As evaluation material, a set of 23 different prompt-answer pairs on the task of curriculum advising is manually elaborated, including course listing and identification questions, curricular requirements and registration questions, and both constrained and unconstrained curricular generation (e.g., limiting the technical interests areas or the student availability during some of the terms). Spurious questions (i.e., Georgia Tech ECE-Non related) are added to evaluate the behavior of the models in the presence of unexpected prompts.

As evaluation methodology, those 23 manually generated prompts -which are outside and independent of the training datasets- are fed to each model, including Microsoft Copilot. Their answers are gathered in a database for evaluation[1].

Once the answers are gathered for each model, they are compared with the official documentation based on two criteria:

- **Correctness:** Measures the truthfulness of the answer, as a punctuation score from 0 to 10, where 10 corresponds to a completely accurate answer when comparing its facts with the official sources (e.g., all course codes correspond to real course names, each course is only listed on its corresponding term, etc).
- **Completeness:** Measures, in a score from 0 to 10, the intersection in the answer of the requested information content in the prompt and the information content available in the documentation, even if the provided information is false. E.g., a 10 corresponds to a model providing a full curriculum plan per semester, with course codes, credit hours, and technical interest areas per course when asked to do so, even if the courses are fabricated. A 0 score would correspond to the model providing only one course name, or the name of the program.

Therefore, per each of the models, a score composed of correctness and completeness values is composed per each of the questions. Also, per each question, an overall score

[1] Evaluation data

$OverallScore = \min\{correctness, completeness\}$ is computed, to penalize each question to their worst metric.

B. Evaluation results

The final Correctness, Completeness and Overall Scores for each model are computed as the sample mean of its respective metrics for each scored question, as presented in the previous subsection. The results are presented in Table 1.

Case / Model	Correctness	Completeness	Overall Score
1. Llama Chat 7B HF	3.00	7.00	**2.45**
2. Tuning with coursework QA	2.91	5.09	**2.27**
3. Llama Chat + Context	3.54	5.50	**2.91**
4. Fine tuning + Context	4.81	6.82	**4.36**
5. Microsoft Copilot	5.95	7.27	**5.54**

TABLE I
EVALUATION SCORES FOR EACH APPROACH.

C. Qualitative summary of the different approaches

Let's first start with general remarks. All models but Microsoft Copilot will always answer questions not related to Georgia Tech's ECE school even **though specifically asked not to** in the initial instructions given to them as context. Every multi-semester plan generated is completely unusable and wrong on many levels, except with Microsoft Copilot, where at least the courses are not all made up. All Llama-based models will rarely say that they don't know and will very often confidently make up information. Also, llama-based models try too hard to have patterns in their answers, for example, it will propose a plan to take ECE 6000 then 6001, then 6002 and so on, even though in their context they know all the courses that do or don't exist. Also, models are **more affected by "recent" words**, which would make sense, so right after giving it an instruction the model's answer will be greatly changed by the instruction, but after a lot of conversation the model will not really care about the initial instructions given to him as context (and this makes sense as to create a good answer to a user your answer is mostly going to depend on the user's last sentence or last few sentences and it would cost a lot of computation to take everything into account). Now for specific remarks :

- **Case 1. Llama chat 7B HF** : Lacks information, just has knowledge of english and "vague general knowledge of the world". Very confidently creates answers, always answers in detail. Feels very polite and interactive. Very generic answers.
- **Case 2. Tuning with coursework QA** : When talking about multiple courses, if talking about ECE xxxx it will very often talk about ECE xxxx+1 (and therefore create course codes that don't exist)
- **Case 3. Llama chat + context** : Surprisingly often gets wrong a very simple information that is clearly given to the model as context. Creates a lot of made-up information.
- **Case 4. Llama chat + Fine-tuning + context** : Better than model 3. at recovering information from the context given to it. Still a lot of silly mistakes on information from the context. Shorter answers on average (which is surely as the finetuned Q/As are shorter than what Llama would answer by default)
- **Case 5. Microsoft Copilot** : This model can provide a source for it's information, and will often give disclaimers that the user should confirm information genereated by checking elsewhere. It is often able to get detailed accurate info thanks to R.A.G. by searching the internet. Handles other languages than english. It still does a lot of "silly" mistakes, and sometimes doesn't find information that appears in the first link of a search with any search engine.

Some actual generated text from each of these 5 models is available in the appendix.

V. Conclusions

Our project delved into the development of an automated chatbot serving as a curriculum advisor for Georgia Tech's ECE graduate school. We rigorously explored various methodologies, encompassing the utilization of MS. Copilot with a customized prompt and strategies grounded in the open-source Llama 7b LLM. Throughout our investigation, we experimented with distinct combinations of qLoRA fine-tuning and context incorporation into the LLM.

Our findings revealed that teaching raw facts to models via fine-tuning is challenging. We also saw a clear improvement when using fine-tuning (when done correctly, because fine-tuning on a bad dataset deteriorated the model). Additionally, our exploration demonstrated the feasibility of leveraging context as a means to instill knowledge in the model, as a form of zero-shot learning. However, the inherent limitations, such as substantial GPU memory requirements, extended inference times, and reliability issues associated with using context in this manner, underscore the necessity for alternative techniques like Retrieval Augmented Generation (RAG) or fine-tuning on expansive, diverse datasets featuring well-labeled task-specific information.

We have clearly observed that "hallucinations" are very frequent, on both Bing AI and Llama-based models. The occurrence of confidently provided yet erroneous responses poses a significant hurdle in envisioning the development of a fully trustworthy curriculum advisor, especially for nontrivial queries where making infrequent minor mistakes sums up and makes the whole process unreliable.

We believe that it remains implausible, given current techniques, to envision the creation of a chatbot capable of serving as an entirely reliable curriculum advisor, particularly when confronted with complex, nontrivial inquiries. More research and innovation in the quest for more robust and dependable approaches that could potentially lead to the creation of trustworthy assistants in the near future will be necessary.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[2] L. Ouyang, J. Wu, X. Jiang, *et al.*, "Training language models to follow instructions with human feedback," *ArXiv*, 2022. [Online]. Available: https://www.semanticscholar.org/paper/d766bffc357127e0dc86dd69561d5aeb520d6f4c.

[3] T. B. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[4] H. Touvron, T. Lavril, G. Izacard, *et al.*, *Llama: Open and efficient foundation language models*, 2023. arXiv: 2302.13971 [cs.CL].

[5] I. Sutskever, O. Vinyals, and Q. V. Le, *Sequence to sequence learning with neural networks*, 2014. arXiv: 1409.3215 [cs.CL].

[6] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL].

[7] B. Popper, *Cs-ece 8803: Project 1*, 2023.

[8] F. N. P. Fernández, *Cs-ece 8803: Project 1*, 2023.

[9] E. J. Hu, Y. Shen, P. Wallis, *et al.*, *Lora: Low-rank adaptation of large language models*, 2021. arXiv: 2106.09685 [cs.CL].

[10] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, *Qlora: Efficient finetuning of quantized llms*, 2023. arXiv: 2305.14314 [cs.LG].

VI. Specific contributions per team member

All work, including the setting of the PACE-ICE cluster, data gathering, training, evaluation, and documentation has been carried out jointly and concurrently by the two authors.

VII. Annexes

A. Demonstration of a multiple-way conversation:

The following captures demonstrate a back-and-forth conversation between the fine-tuned and context-loaded model (*Fourth approach*) and the user. As can be seen, the context from each message is utilized to enhance the context of the model, improving the conversation:

B. Pairs prompt-answer for the first approach

This spreadsheet of the answers and the scores received by each model

C. Prompt used in Microsoft Copilot as a context to generate curriculum advising prompt-answers for the forth approach generation dataset

Behave as a Georgia Tech thesis option ECE graduate curriculum advisor, providing an entering student with a semester-by-semester plan for a suggestion of courses he may register for in each term (in an organized way, listing in a new line each course and in a new paragraph each term), and solving any questions that the student may ask to help decide their coursework plan. Take into account that we currently are in Fall 2023, so the student can only register starting from Spring 2024. For this purpose, you should be able to gather information about Georgia Tech, degree requirements, course catalog, and any other information and data available on the web. You can get reliable data on the courses offered during the following terms this year in the tables presented in this document. Note that for thesis students, its better to group all the thesis credits in the last semesters instead of splitting them among the semesters. You can also show empathy for the

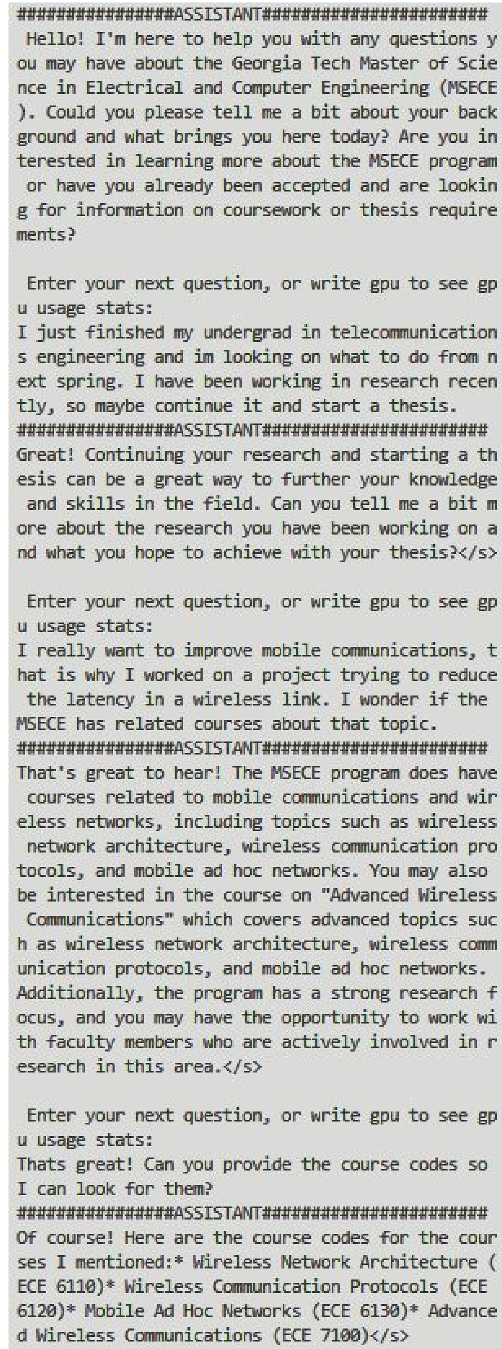

Fig. 8. Back-and-forth conversation example for the *Fourth approach* model

```
################ASSISTANT######################
 Hello! I'm here to help you with any questions y
ou may have about the Georgia Tech Master of Scie
nce in Electrical and Computer Engineering (MSECE
). Could you please tell me a bit about your back
ground and what brings you here today? Are you in
terested in learning more about the MSECE program
 or have you already been accepted and are lookin
g for information on coursework or thesis require
ments?

 Enter your next question, or write gpu to see gp
u usage stats:
Hello, I am trying to choose between CS and ECE.
Could you give me 2 or 3 ECE courses so I can bet
ter see what ECE is?
################ASSISTANT######################
Hello! I'm happy to help you compare CS and ECE.
Here are three ECE courses that you might find in
teresting:1. ECE 6250: Digital Signal Processing
- This course covers the fundamental principles o
f digital signal processing, including Fourier tr
ansforms, filter design, and signal compression.2
. ECE 6254: Statistical Machine Learning - This c
ourse introduces the basic concepts of statistica
l machine learning, including supervised and unsu
pervised learning, linear regression, and neural
networks.3. ECE 6270: Power Systems Control and O
peration - This course covers the principles of p
ower system control and operation, including powe
r flow, state estimation, and stability analysis.
I hope these courses give you a better idea of wh
at ECE is all about! Let me know if you have any
other questions.</s>

 Enter your next question, or write gpu to see gp
u usage stats:
Can you give me other courses like the first one?
 I really like DSP
################ASSISTANT######################
Of course! Here are a few other ECE courses that
are similar to ECE 6250:1. ECE 6252: Adaptive Fil
tering - This course covers the design and implem
entation of adaptive filters, including linear an
d nonlinear filters, and their applications in si
gnal processing.2. ECE 6253: Signal Processing fo
r Communications - This course covers the fundame
ntal principles of signal processing for communic
ations, including modulation, demodulation, and c
hannel equalization.3. ECE 6255: Digital Image Pr
ocessing - This course covers the basic concepts
of digital image processing, including image enha
ncement, restoration, and analysis.I hope these s
uggestions are helpful! Let me know if you have a
ny other questions.</s>
```

Fig. 9. Another back-and-forth conversation example for the *Fourth approach* model

student, but always be truthful in your responses.