TECH-GB.3332 Fall 2024 Prof. Alex Tuzhilin

**Assignment 3**

**Due date:** November 6

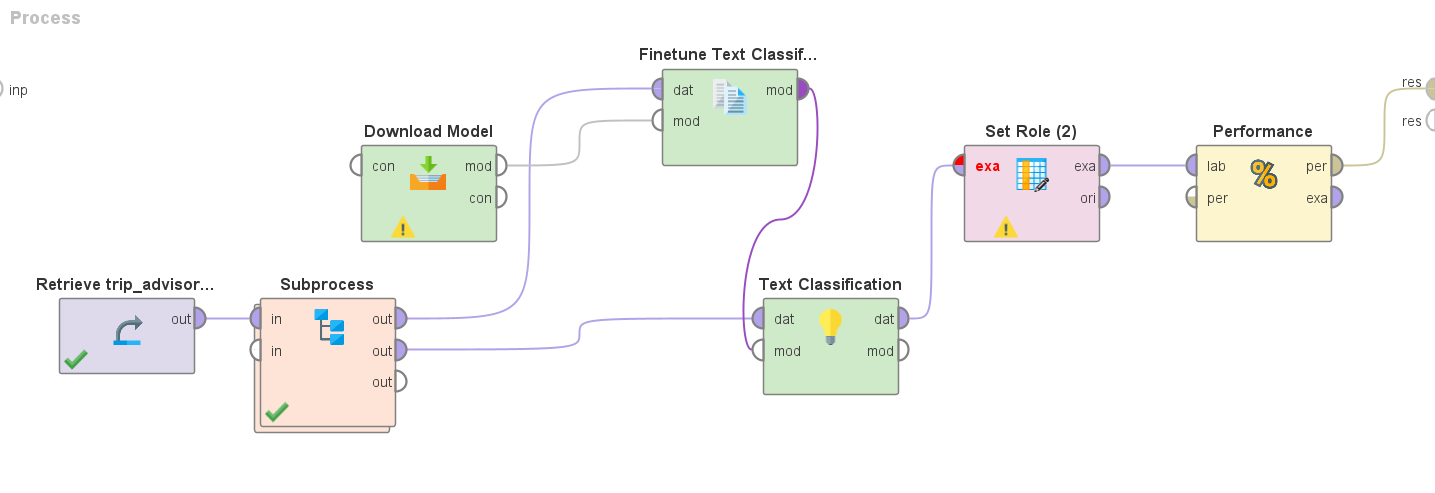
# **Part 1. Fine-tuning a Pre-trained Generative AI Model**

To start this assignment, download the [trip\_advisor\_2000.xlsx](https://docs.google.com/spreadsheets/d/1sw2LqJ_AFfzyFlKwMJFsMF9shhV3enfm/edit?usp=sharing&ouid=114141838929637485833&rtpof=true&sd=true) dataset and open AI Studio. Then, import the dataset into AI Studio following the same process we used in the Lab session. Specifically, set “review\_rating” as polynominal or nominal instead of integers.

***Finetuning Pre-trained Models on Ratings.*** In this assignment, you will take a pre-trained mode “bert-base-multilingual-uncased-sentiment” from Hugging Face that is trained for sentiment analysis on product review in six languages, such as English, Dutch, German, French, Spanish and Italian. This model predicts the review sentiment as a number of stars (1 to 5 stars) as it was done in Part 4 (the “Generative Models” part) of the Lab. Then, you will subsequently finetune this pre-trained model on the Trip Advisor reviews to predict customer ratings (as a number from 1 to 5) by using the AI Studio Generative Model Extensions.

To accomplish this task, follow the fine-tuning process presented in Figure 1. As a starting point, drag the original “trip\_advisor\_2000” dataset (that you just imported into the AI Studio as discussed in the first paragraph) to the Process panel. Make sure that your “review\_rating” column is of type *nominal/polynominal*. If not, change it as was done in Assignment 2.

In the Operators panel, search one-by-one for the “Subprocess”, “Download Model”, “Finetune Text Classification”, “Text Classification”, “Set Role” and “Performance (Classification)” operators and drag them to the Process panel (as we have done it in the Lab and Assignment 2). Connect all the selected operators as shown in Figure 1.



**Figure 1.** The Flowchart of Fine-tuning Model.

After that, set the parameters of all these operators as follows:

*Subprocess:* Click on the “Subprocess” operator and go to the “Subprocess” panel, where you search for the “Set Role” and the “Split Data” operators in the Operators panel and then drag them to the subprocess panel (similar to how you did it in Figure 2 of Assignment 2). Note that the “Split Data” operator splits the dataset into the train and the test sets in the 0.8/0.2 ratio (80% training and 20% testing data, as was done in the Lab and Assignment 2) and the “Set Role” operator sets the “review\_rating” field as label. [Note that the Subprocess operator is implemented similar to the way it was done in Figure 2 from Assignment 2, except that there is no “Select Attributes” component here.]

*Download Model*: As explained earlier, the Download model downloads the pre-trained model from Hugging Face into AI studio for the subsequent work (finetuning it). To set up this downloaded model, specify its parameters in the Parameters panel as follows: model = “nlptown/bert-base-multilingual-uncased-sentiment”; type = “text classification.” Finally, set the “project folder” parameter by clicking on the appropriate location of your choice in the Local Repository directory.

*Finetune Text Classification*: Finetune Text Classification Operator will finetune the pretrained model “nlptown/bert-base-multilingual-uncased-sentiment” on the trip advisor dataset for text classification tasks. Set the parameters of this model similar to what you have just done for the Download Model: specify the storage type and the project folder according to your preferences; then set input column = review\_content; target column = review\_rating; epoch = 5; device = either GPU or CPU options according to your laptop hardware configuration (and what you specified during the AI Studio installation process).

*Text Classification*: Set the parameters of this operator as follows: check the box of use local model; name = “pred\_rating”; prompt = “[[review\_content]]”; device = either GPU or CPU options according to your laptop hardware.

*Set Role:* Specify the following two (attribute name, target role) pairs: (review\_rating, label) and (pred\_rating, prediction) [click on the “add entry” button for each pair]. For example, with (review\_rating, label) pair, you will select “review\_rating” on the left attribute name and “label” as target role. Note that “pred\_rating” attribute name needs your manual entry/typing because the “pred\_rating” column was not created before executing the program).

*Performance (Classification)*: select “accuracy” as the performance metric.

Finally, click the “Run” button to run the fine-tuning process. Note that this may take some time (can be up to 1 hour – as we discussed, some of the AI problems can be computationally demanding!). So, be patient!

Report the results and the running time for this fine-tuned model. Please comment on the differences between the results produced by the fine-tuned model you just built in this assignment and the pre-trained model produced in Part 4 (the “Generative Models” part) of the Lab. Also, how good are the accuracy performance results of these models, as compared with the results produced in Part 3 of Assignment 2 (i.e. are they better or worse than those other results)? Please, justify your answer.

# 

# **Part 2. Prompting Techniques for Generative Models**

In this part, you will learn to invoke ChatGPT using API (from within AI Studio). You will also learn how to use Zero-Shot and Few-Shot methods in this assignment. As a first step, we connect to the OpenAI platform as follows.

1. ***Connection to the OpenAI Platform***

(a) *Logging in, Setting an Account and Creating the Secret Key.* Go to the OpenAI API website <https://platform.openai.com/docs/overview>. Log in with your existing account or create an account if you don’t have one. After logging in, click on “Dashboard” at the top right. Click on the “API keys” link located in the lower-left part of the panel. If you haven’t verified your phone number, click on the “start verification” process (*If your account is new and your phone has never been used for the Openai account verification, you should receive a $5 free trail. If there is no free trail, then you would have to pay*). Otherwise, click on the Create new secret key at the top right of the panel. In the popup window, enter “AI Studio” for Name and click on “Create secret key”. Save the key to somewhere else for future use.

(b) *Paying for Using the OpenAI Platform*. Unfortunately, OpenAI wants you to pay for using their platform and their API. The good thing is that you can pay only $5, and this will allow you to use the Openai platform for a long time, unless you significantly scale its usage. Moreover, I’ll be happy to reimburse you for paying $5 to OpenAI if you want/need it. To add $5 to your OpenAI account, click on the setting icon at the top right of the screen. Then click on the “Billing” option on the left panel. Click on “Add payment” and make a payment to Openai. (e.g., $5, as we discussed before).

(c) *Connecting the AI Studio with the OpenAI Platform*. Next, go to the AI Studio and click on the “Connections” tab on the top. Then click on the “Create Connection” button. In the “Create a new connection” window, select “Dictionary” for the Connection Type, type “openai” in the “Connection Name” field, and then click on the “Create” button. Then click on the “Add entry” button and enter “api\_key” on the left and paste your openai *key* on the right. As a result, you established a connection between the AI Studio and the OpenAI platform.

1. ***LLM as Zero-Shot Learning Algorithm***

In the “Repository” panel, go to the “Local Repository” directory and then into its “Connections” subdirectory. Grab “openai” from the Connections subdirectory and drag it to the “Process” panel. Then do the same with the “trip\_advisor\_2000” dataset: also grab and drag it to the Process panel. As a result, the two operators will appear on the left, as is shown in Figure 2. Then in the “Operators” panel, search and drag “Sample”, “Send Prompt (OpenAI)”, “Set Role”, and “Performance (Classification)” operators to the “Process” Panel and link them as shown in Figure 2. Also, set the parameters of these operators as follows:

*Sample*: sample size = 100; check the “use local random seed” field; local random seed = 1992.

*Send Prompt (OpenAI)*: model = gpt-3.5-turbo; name= rating\_pred; temperature = 0.0. Then click on the prompt and enter the following:

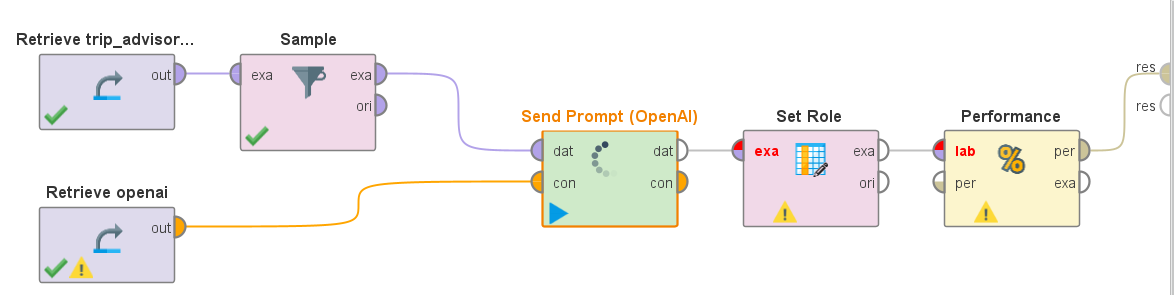
*Please rate the restaurant from 1 to 5 based on the following review. 1 for an extremely bad restaurant and 5 for an extremely good restaurant. Please, reply directly with only the integer rating.*

*Review: [[review\_content]]*

*Rate:*

*Set Role*: Set “review\_rating” as label, and “rating\_pred” as prediction.

*Performance*: check “accuracy” as the performance metric.



**Figure 2. OpenAI Prompts with Trip Advisor Restaurant Reviews.**

Then run this model and report the performance results and the running time. This process may take several minutes. Also, briefly comment on the performance results, i.e. how is the performance compared to the finetuning models in Part 1. What are the potential reasons for such differences?

1. ***LLM as Few-Shot Learning Algorithm***

Use the same process as in Figure 2. However, for the “Send Prompt (OpenAI)” parameter, click on the prompt and edit it as follows:

*Please rate the restaurant from 1 to 5 based on the following review. 1 for an extremely bad restaurant and 5 for an extremely good restaurant. Please, reply directly with only the integer rating.*

*Review 1: Went to the Punch Tavern on the evening of 05/02/15 to celebrate my birthday. The bar staff were very rude and extremely unhelpful. This was experienced by a number of people in my party. I was considering booking the venue for our christmas party, but based on last nights service i will be looking elsewhere*

*Rate: 1*

*Review 2: great service, friendly staff, amazing food, this place never disappoints. Opted for the taster menu.*

*Rate: 5*

*Review 3: The menu was limited but what was on offer was very good. It's a fun place, busy quite noisy definitely go again for lunch or dinner*

*Rate: 4*

*Review 4: Went to the Ivy to a corporate dinner so was in the private rooms. The food was great and te service attentive but i expected more given all the good things i've heard about it. The room have charm but felt a bit run down.*

*Rate: 3*

*Review 5: Went here for valentines dinner. Set menu which all looked good. Got placed on a table which is something you find on a holiday terrace in Greece. Got 'plonked' a glass of champagne with no ask of any other option of drink. Food was good. Not great though. My prawn starters were not cooked but replacement was prompt. The cod dish was advertised with capers but there were none. But was decent. My wife had steak which was good. Creme brûlée was not good. Service was patchy but obviously over worked that night, Not the price tag for that experience, that's all.*

*Rate: 2*

*Review: [[review\_content]]*

*Rate:*

Note that, unlike the previous Zero-Shot LLM model (that had only the prompt without any additional review examples), this current “a few-shot” LLM model has five examples of reviews together with the corresponding ratings (a “few-shots,” as these examples are known in GenAI and ML in general). Also note that these examples are used by the LLM to get additional training to deliver more “informed” results.

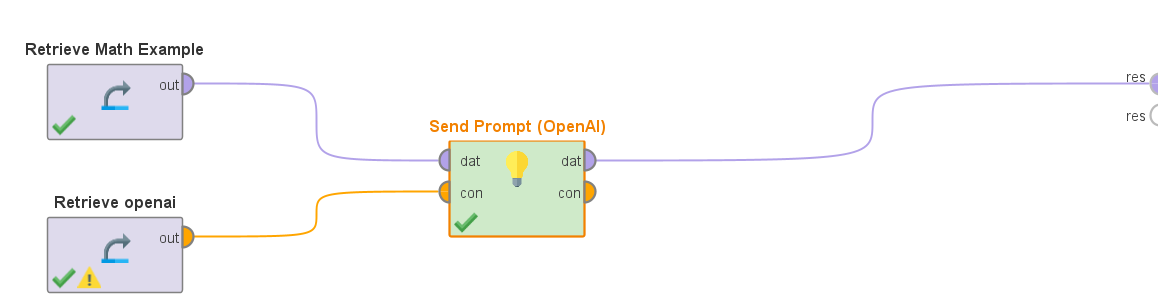
After entering the prompt, run this model and report the performance results and the running time. Also, briefly comment on the performance results. How would these performance results compare with those achieved by the Zero-shot model from Step 2 of Part 2? How would you explain the performance differences?

# **Part 3. Prompt Engineering**

In this part, we will apply the well-established prompt engineering technique “Chain-of-Thought (CoT)” to solving some mathematical problems. See the paper “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models” published by Google researchers (<https://arxiv.org/pdf/2201.11903>) for the details and specifics of this method.

**(a) ChatGPT Solving Mathematical Problems (without using Chain-of-Thought)**

Download [Math Example](https://docs.google.com/spreadsheets/d/1D5U19GIQ0pq_uzhWzvKDR4TrZjq9cUQdQA8EDM_TBuk/edit?usp=sharing) onto your laptop and open a blank process in the AI Studio. Then import the downloaded “Math Example” dataset into the Local Repository. From the “Repository” panel, drag “openai” from the “Local Repository\Connections” directory, as it was done in Part 2, Step 2. Also drag and drop the “Math Example” Dataset into the “Process” Panel. Then in the “Operators” panel, search and drag “Send Prompt (OpenAI)” to the “Process” Panel. Finally, link all the three operators, as is shown in Figure 3, and set the parameters as follows:



**Figure 3. ChatGPT Solving Mathematical Problems.**

*Send Prompt (OpenAI)*: Set model = gpt-3.5-turbo; name=openai; and temperature = 0.0. Then click on the prompt and enter the following:

*Stop using the chain-of-thought. Answer the following question:*

*Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?*

*A: 11*

*Q: [[Question]]*

*A:*

Run this model and report how many questions ChatGPT answered correctly. Also, briefly comment on the performance results [note that the answer for each question is shown in the dataset in the “Answer” column].

**(b) ChatGPT Solving Mathematical Problems Using Chain-of-Thought Prompting**

Modify the prompt in the “Send Prompt (OpenAI)” operator (see Figure 3) as follows:

*Answer the following question:*

*Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?*

*A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.*

*Q: [[Question]]*

*A:*

Run this model and report how many questions ChatGPT answered correctly this time. Also, briefly comment on the performance results, i.e. how is the performance compared with the results from Part 3(a) (where the CoT prompt engineering technique was not used)? How would you explain the differences?

# **Deliverables:**

Your submission file should contain two parts:

1. A pdf report that contains:
   1. Model Design (flowcharts similar to the examples shown in the figures above)
   2. Performance outputs, as requested in each question.
   3. Your comments on the model performance, as asked in each question.
2. A replicability folder in the .zip format that contains:
   1. The AI studio processes. To collect them, click on the File menu at the top left corner of AI Studio and then click on the Export Process option. Export your process file in the .rmp format as follows:
      1. Save the finetuning Process in Part 1as Finetuning.rmp
      2. Save the zero-shot openai Process in Part 2-2 as Zero-shot.rmp
      3. Save the few-shot openai Process in Part 2-3 as Few-shot.rmp
      4. Save the math Process in Part 3-(a) as math.rmp
      5. Save the math Process with CoT in Part 3-(b) as math\_cot.rmp