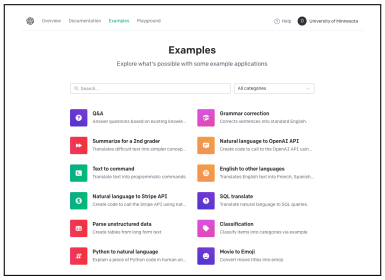
The computational trend of NLP research is shifting from feature engineering to representation learning to pretraining-finetuning to ver recently prompt engineering. Pre-trained language models (or other generative models for other modalities) allow the extraction of diverse and intrinsic knowledge from human-written texts/images/videos and their pairs. This assignment requires you to **explore the limits and capabilities of pre-trained large language models** by designing your own prompts, observing their outputs, and understanding their shortcomings. This homework assumes that you fully understand the course material on prompting[*] taught on October 27. Please follow the three steps below and submit your prompt spreadsheet and report.

Step 1: Getting used to OpenAI's GPT-3 Playground and APIs

Please make your account in OpenAI's playground: https://beta.openai.com/playground. During your first three months of making a free account, you can basically use $18 worth of free experiments. For example, if you use Davinci[†] model ($0.02 per 1K tokens), you can use a total of 900,000 tokens to generate in free.



Examples

Explore what's possible with some example applications

Q&A — Answer questions based on existing knowle...

Grammar correction — Corrects sentences into standard English.

Summarize for a 2nd grader — Translates difficult text into simpler concep...

Natural language to OpenAI API — Create code to call to the OpenAI API usin...

Text to command — Translate text into programmatic commands.

English to other languages — Translates English text into French, Spanish...

Natural language to Stripe API — Create code to call the Stripe API using nat...

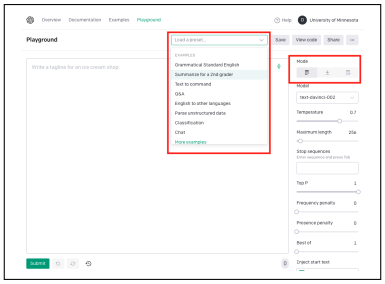
SQL translate — Translate natural language to SQL queries.

Parse unstructured data — Create tables from long form text

Classification — Classify items into categories via example.

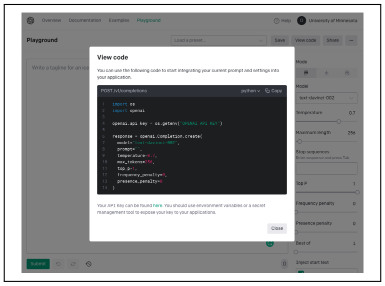
Python to natural language — Explain a piece of Python code in human un...

Movie to Emoji — Convert movie titles into emoji.



Write a tagline for an ice cream shop.

Model: text-davinci-002
Temperature: 0.7
Maximum length: 256
Stop sequences (Enter sequence and press Tab)
Top P: 1
Frequency penalty: 0
Presence penalty: 0
Best of: 1
Inject start text



View code

You can use the following code to start integrating your current prompt and settings into your application.

POST /v1/completions

```python
import os
import openai

openai.api_key = os.getenv("OPENAI_API_KEY")

response = openai.Completion.create(
  model="text-davinci-002",
  prompt="",
  temperature=0.7,
  max_tokens=256,
  top_p=1,
  frequency_penalty=0,
  presence_penalty=0
)
```

Your API Key can be found here. You should use environment variables or a secret management tool to expose your key to your applications.

Please take a look at existing example tasks in Examples tab[‡], such as Queston Answering, Summarization, and Text-to-Command, and load preset prompts in Playground tab[§], as described in the figures above. You can also choose three types of modes, *Complete*, *Insert*, and *Edit* in the Mode tab on right panel (red rectangle box on the right figure above). You can choose one of these modes for every prompt or choose two different modes half and half.

You task is to make a pair of single prompt and expected output, (Prompt, Expected Answer), and report the output from GPT3. For example, my prompt-answer pair is ("Task: is this text positive or negative? \n Input: Today's weather is sunny and great! \n Output: ", Positive) and the actual output from GPT3 is (Negative), which is not matched with my expected answer. You are not allowed to use any examples in the Example tab or preset prompts. Be creative!

Step 2: Understand Current Prompting Techniques.

This assignment requires you to understand the state-of-the-art prompting techniques. Refer to the following articles and papers (you can find links to the original articles in the Reference section):

- Discrete prompting methods: Auto-prompting method [SRLI+20] and analysis on the limitation and tricks in discrete prompting [ZWF+21]
- Soft prompting methods: Prefix-tuning [LL21] and Prompt-tuning [LARC21].
- Stress test of GPT3 on various aspects: commonsense reasoning [MD20], hypes and ethics [BGMMS21], and planning [VOSK22].

[*] https://dykang.github.io/classes/csci5980/Fall2022/

[†] https://openai.com/api/pricing/

[‡] https://beta.openai.com/examples/

[§] https://beta.openai.com/playground

- Some prompting tricks like Chain-of-thoughts [WWS+22] and applications to human-GPT3 collaboration for text editing [SDYJ+22] and poetry writing [CPH22].

Your task is to choose one bullet point from the list above and summarize the papers and limitations of current methods. (maximum 2 pages; don't forget to properly cite them in your report)

Step 3: Prompting Yourself

The last step involves designing your own (discrete) prompts! Also, it's time to get creative! Your goal is to *fool the GPT3 system* by finding adversarial examples by yourself and provide reasonable justification of the failure cases. The first step is to **pick three of the following aspects you would like to explore**. You can also think of **your own aspect to explore** but please describe it clearly in the report:

- **Creativity**, e.g., writing next possible sentences in your own story prompt
- **Generalization** to unseen, novel situations or tasks, e.g., design a totally unseen, new task you can get benefit from GPT3 prompting.
- **Grammatical errors, typo**, and other language fluency measurement: e.g., GPT-3 makes mistakes grammar/different tense with popular phrases and prefers more active speech?
- **Memorization** of existing facts and **Commonsense Knowledge**: e.g., GPT-3 memorizes the phone number of White House? GPT-3 is able to understand Newton's laws of motion?
- **Biases and Ethical Concerns**: e.g., GPT-3 prefers to set higher salaries for men than women? GPT-3 generates more offensive languages toward Asian people than white people?
- **Temporal/Spatial reasoning**: e.g., GPT-3 is not able to understand temporal and spatial state transition of objects in real world?
- **Mathematical reasoning**: e.g., GPT-3 is able to solve Fibonacci sequence?
- **Reasoning on Commonsense, Morality, and Legality**: e.g., GPT-3 can address some commonsense or social issues like morality and legality?
- **Applications on NLP tasks**: e.g., GPT-3 translates low-resource languages? GPT-3 can write code for your assignment? GPT-3 can summarize/rewrite your emails?

You should make **three adversarial task prompts (Task Description ; Zero/Few Examples ; Input Task, Expected Answer)** for each ask you choose above, where tasks in each aspect are different from each other. Note that you are finding **adversarial** cases in which GPT-3 outputs don't match your expected, predicted results. Each task prompt should also be tested with a zero-shot setup (no example given) and a few-shot setup (one or three examples given). In total, you need to create at least **three aspects** x **three tasks** x **two setup; zero/few = 18 adversarial prompts**. You can of course try as many prompts as you wish and there would be bonus points based on the quality/creativity of your prompts (See the prompt evaluation criteria below). After prompting, you should provide reasonable reasons for failures and possible ways to improve them **justification of those failures**.

Here are some notes and tips for your prompt design:

- NOTE: You have to provide a reasonable quality of task description and examples in your prompts, and make sure that GPT-3's failure does not come from the quality of your prompt design, but is mainly caused by the lack of inherent capabilities of GPT-3. You can find high-quality prompts through trial-and-error with GPT-3 in Playground or in your python code using OpenAI's API (See View Code tab in Playground interface in the figure above).
- We are Scientists! Try different task descriptions and prompt examples, and see if GPT-3 always fails deterministically.

- Once again, you cannot use examples from the Example tab, predefined prompts, or previous papers. It will be treated as *cheating* if I find the same prompt used before.
- Here are some additional tips you may consider during prompting:
  - Find novel tasks you/GPT3 can't do.
  - Find tasks that GPT3 can do a better job than humans.
  - Find cases where even humans do not agree on each other and seeing how GPT3 can handle this human disagreement
  - Find unseen (probably not seen in the training data) but realistic cases
  - Find unseen and unrealistic cases

Deliverable

Please upload your spreadsheet of your designed prompts and report to Canvas by Nov 17, 11:59pm.

Spreadsheet: Your designed prompts should be contained in a spreadsheet (e.g., CVS, Excel files, Google Spreadsheet). In each row, you can write a prompt, and in your columns, you can include the following information:

- Mode (Complete, Edit, or/and Insert)
- Aspect (e.g., Creativity, Generalization)
- Task Description
- Instructions (if you used Edit mode)
- Number of examples
- Examples
- Input Task (without answer)
- Expected Answer
- Predicted Answer by GPT3
- Setting (e.g., Engine - text-davinci-002 Temperature - 0 (deterministic) Max length - 256 Stop sequences - none Top P - 1 Frequency penalty - 0 Presence penalty - 0 Best of - 1")
- Justification of failure
- Takeaways or other interesting things you learn

Your designed prompts will be evaluated based on the following criteria[¶] and you will receive extra bonus point if you win one of these awards.

- Best Research Application: Discover a finding that may lead to further research or publication
- Best Misuse Case: Discover a scenario where GPT-3 generates harmful content in a way that might be difficult for GPT-3 admins to catch Ideally, also determine a way one might go about detecting such harmful content
- Best Mistake Case: Discover an interesting case where GPT-3 fails by not giving you the output you want or expect
- Best Prompt: Discover an interesting prompt
- Persistence Award: Trying to get something to work over and over again, even if you aren't ultimately successful
- Arbitrary Finding: Find something completely random but interesting

Report: Maximum four pages PDF. Your report needs to include the following content in each step:

- Step 1: What mode of GPT3 did you choose (Complete, Insert, Edit)? Please show me your example prompt-answer pair and GPT3 output.
- Step 2: Which papers did you choose to read? (Please cite them properly) Summarize the papers and limitations of the current methods.

[¶] Borrowed these criteria from London Lowmanstone's GPT-3 Edit/Insert Hackathon, March, 2022. Thanks, London!

- Step 3: Explain why you chose the aspects and how your prompts were designed. Discuss challenges you encountered during your homework and your general thoughts on language model prompting.

References

[BGMMS21] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery. https://doi.org/10.1145/3442188.3445922.

[CPH22] Tuhin Chakrabarty, Vishakh Padmakumar, and He He. Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing. *arXiv preprint arXiv:2210.13669*, 2022. https://arxiv.org/abs/2210.13669.

[LARC21] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. https://aclanthology.org/2021.emnlp-main.243.

[LL21] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, August 2021. Association for Computational Linguistics. https://aclanthology.org/2021.acl-long.353.

[MD20] Gary Marcus and Ernest Davis. Experiments testing gpt-3's ability at commonsense reasoning: results, 2020. https://cs.nyu.edu/~davise/papers/GPT3CompleteTests.html.

[SDYJ+22] Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. Peer: A collaborative language model. *arXiv preprint arXiv:2208.11663*, 2022. https://arxiv.org/abs/2208.11663.

[SRLI+20] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online, November 2020. Association for Computational Linguistics. https://aclanthology.org/2020.emnlp-main.346.

[VOSK22] Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). *arXiv preprint arXiv:2206.10498*, 2022. https://arxiv.org/pdf/2206.10498.pdf.

[WWS+22] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models, 2022. https://arxiv.org/abs/2201.11903.

[ZWF+21] Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models, 2021. https://arxiv.org/abs/2102.09690.