CSC6203/CIE6021:
Large Language Model

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Tutorial1: How to automatically use ChatGPT in a batch?

Winter 2023

Benyou Wang
School of Data Science

Let’s briefly review what we have learned

Lecture1: What & Why LLM in Big picture & ChatGPT

- **Large language models?**
- **Introduction to ChatGPT**

What?

Sentence: “the cat sat on the mat”

$$P(\text{the cat sat on the mat}) = P(\text{the}) * P(\text{cat}|\text{the}) * P(\text{sat}|\text{the cat}) * P(\text{on}|\text{the cat sat}) * P(\text{the}|\text{the cat sat on}) * P(\text{mat}|\text{the cat sat on the})$$

Implicit order

Why?

Why Larger language models

- More world **knowledge** (LAMA)
  - Language models as knowledge base?
- Larger capacity to learn problem-solving **Abilities**
  - Coding, revising articles, reasoning etc.
- Better **generalization** to unseen tasks
- **Emergent** ability (涌现能力)

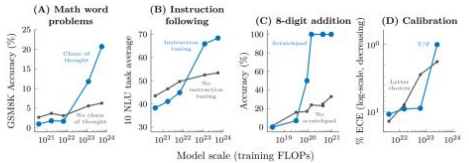


Figure 3: Specialized prompting or finetuning methods can be emergent in that they do not have a positive effect until a certain model scale. A: Wei et al. (2022b). B: Wei et al. (2022a). C: Nye et al. (2021). D: Kadavath et al. (2022). An analogous figure with number of parameters on the x-axis instead of training FLOPs is given in Figure 12. The model shown in A-C is LaMDA (Thoppilan et al., 2022), and the model shown in D is from Anthropic.

Jared Kaplan et. al **Scaling Laws for Neural Language Models**
Jason Wei et. Al. Emergent Abilities of Large Language Models.

Lecture2: Language modeling in details and beyond

- **Language model in a narrow sense**
  (Probability theory, N-gram language model)
- Language model in broad sense

The meaning of a word:

Representing words by their context

Distributional semantics: **A word's meaning is given by the words that frequently appear close-by**

- "*You shall know a word by the company it keeps*" (J. R. Firth 1957: 11)
- **One of the most successful ideas of modern statistical NLP!**

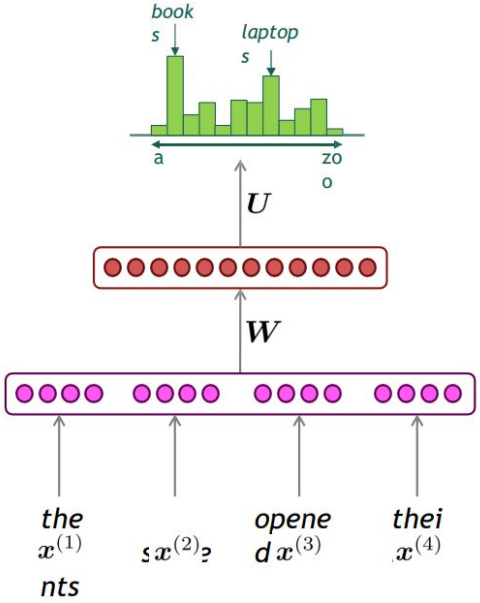
• When a word *w* appears in a text, its context is the set of words that appear nearby (within a fixed-size window).
• We use the many contexts of w to build up a representation of *w*

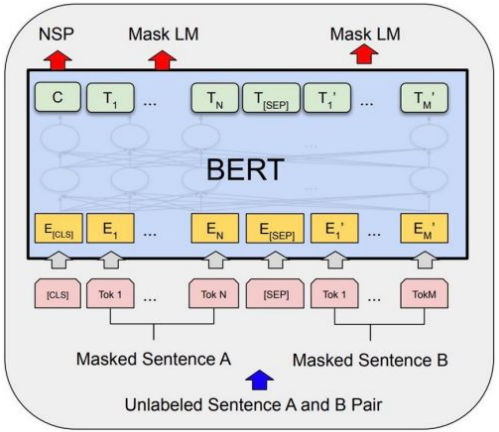
…government debt problems turning into ***banking*** *crises as happened in 2009…*
…saying that Europe needs unified ***banking*** *regulation to replace the hodgepodge…*
…India has just given its ***banking*** *system a shot in the arm…*

These context words will represent ***banking***

https://web.stanford.edu/class/cs224n/slides/cs224n-2023-lecture01-wordvecs1.pdf

The meaning of word sequence:

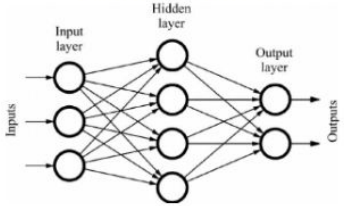




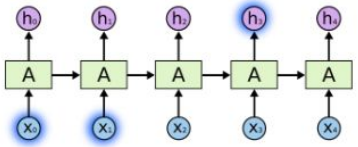
Lecture3: Architecture in details and beyond

Sequence modeling

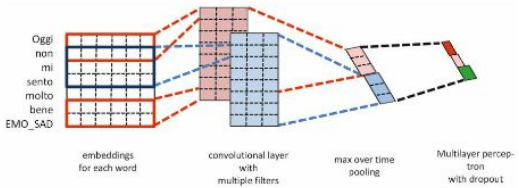
Feed-forward NNs



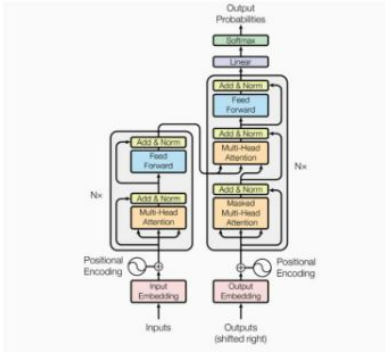
Recurrent NNs



Convolutional NNs



Transformer



SAN(Self-Attention):
Token-level interaction

FFN(Feed-forward):
Feature-level abstraction

The modern deep learning is just using weaker inductive biases and make more data-driven instead of prior-driven.

Now!

Let us enjoy standing on the shoulders of giants

Usage of OpenAI API and Assignment1 Introduction

Today’s Tutorial

- VPN, Keys and basic usage
- Explain decode strategies by explaining Model Parameters in OpenAI API
- Implement Assignment task1 and task2 step by step

VPN, Keys and basic usage

VPN

- Forwarding service of our lab: + openai.api_base = "https://openai.huatuogpt.cn/v1"

```
openai.api_key = OPENAI_API_KEY
openai.api_base = "https://openai.huatuogpt.cn/v1"
response = openai.ChatCompletion.create(
```

- Self-vpn: https://flaff20230820.fastlink-aff02.com/auth/login##

Keys

- 100 keys in gpt3keys.txt
- Exceed the quota: buy here

Prompt Engineering

Dependency

- `Pip install openai`
- `pip install urllib3==1.25.11`

Related resource:

- https://www.promptingguide.ai/zh
- https://github.com/dair-ai/Prompt-Engineering-Guide

Tutorial code link:

```python
import openai

OPENAI_API_KEY =  "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"
openai.api_key =  OPENAI_API_KEY
openai.api_base =  "https://openai.huatuogpt.cn/v1"
response = openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[
    {"role": "system", "content": "You are a helpful assistant."},
    {"role": "user", "content": "Tell me a fun fact."},
    {"role": "assistant", "content": "Sure, here is an interesting fact:"}
  ],
  temperature=0.6,
  max_tokens=30,
  top_p=0.8,
  frequency_penalty=0.6,
  presence_penalty=0.8,
  n=3
)
print(response)
```

Model Parameters in OpenAI API

Model Parameters in OpenAI API

1. Understanding the `openai.ChatCompletion.create()` Function

The strings `"system"`, `"user"`, and `"assistant"` are used to define the role of each message within the conversation.

```
response = openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[
    {"role": "system", "content": "You are a helpful assistant."},
    {"role": "user", "content": "Tell me a fun fact."},
    {"role": "assistant", "content": "Sure, here is an interesting fact:"}
  ],
  temperature=0.6,
  max_tokens=30,
  top_p=0.8,
  frequency_penalty=0.6,
  presence_penalty=0.8
)
```

`"system"`: The "system" role typically provides high-level instructions or context-setting messages.
`"user"`: The "user" role represents the messages or queries from the user.
`"assistant"`: The "assistant" role represents the responses generated by the ChatGPT model.

Model Parameters in OpenAI API

2. Temperature: Adding Randomness to the Responses

A higher value, such as `0.8`, makes the answers more **diverse**, while a lower value, like `0.2`, makes them more **focused** and **deterministic**.

Experience: A value between 0.2 and 0.8 can be effective. Lower values (e.g., 0.2) produce more focused and deterministic responses, while higher values (e.g., 0.8) allow for more randomness.

Why?

Temperature

Scaling randomness: Temperature

- Recall: On timestep t, the model computes a prob distribution P_t by applying the softmax function to a vector of scores $s \in \mathbb{R}^{|V|}$

$$P_t(y_t = w) = \frac{\exp(S_w)}{\sum_{w' \in V} \exp(S_{w'})}$$

- You can apply a *temperature hyperparameter* τ to the softmax to rebalance P_t:

$$P_t(y_t = w) = \frac{\exp(S_w/\tau)}{\sum_{w' \in V} \exp(S_{w'}/\tau)}$$

- Raise the temperature $\tau > 1$: P_t becomes more uniform
  - **More** diverse output (probability is spread around vocab)
- Lower the temperature $\tau < 1$: P_t becomes more spiky
  - **Less** diverse output (probability is concentrated on top words)

CS224N-lec10

Model Parameters in OpenAI API

3. Max Tokens: Limiting the Response Length

The `max_tokens` parameter allows you to limit the length of the generated response. Setting an appropriate value allows you to control the response length and ensure it fits the desired context.

Experience： Default 4096

Model Parameters in OpenAI API

4. Top P (Nucleus Sampling): Controlling Response Quality

Higher values like **0.9** allow more tokens, leading to **diverse** responses, while lower values like **0.2** provide more **focused** and **constrained** answers.

Experience：A value between 0.3 and 0.9 is recommended. Higher values (e.g., 0.9) make the model consider a broader range of possibilities, while lower values (e.g., 0.3) make it more selective.

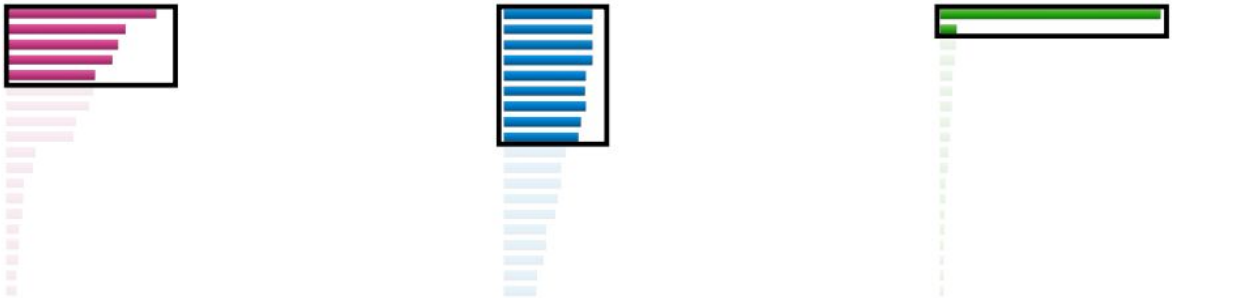
Why?

Top P

Decoding: Top-*p* (nucleus) sampling

- Solution: Top-*p* sampling
  - Sample from all tokens in the top *p* cumulative probability mass (i.e., where mass is concentrated)
  - Varies *k* depending on the uniformity of P_t

$P_t^1(y_t = w \mid \{y\}_{<t})$ $P_t^2(y_t = w \mid \{y\}_{<t})$ $P_t^3(y_t = w \mid \{y\}_{<t})$



CS224N-lec10

Model Parameters in OpenAI API

5. n: Generating Multiple Responses

The `n` parameter allows you to generate multiple alternative completions for a given conversation. By increasing the value of `n`, you can explore different response variations.

Model Parameters in OpenAI API

6. Stop: Customizing stop Conditions

The `stop` parameter allows you to specify a custom condition for the completion. You can provide a string or a list of strings that, when encountered in the generated response, will cause the model to stop generating further tokens. Here's an example

Model Parameters in OpenAI API

7. Frequency Penalty: Controlling Repetitive Responses

It's like telling the computer, "Hey, don't repeat words too much." It stops repetition.

Higher values, like **1.0**, **encourage the model to explore more diverse and novel responses**, while lower values, such as **0.2,** make the **model more likely to repeat information.**

8. Presence Penalty: Controlling Avoidance of Certain Topics

It's like saying, "Hey, use a variety of words, not just the same ones." It encourages variety.

Higher values, such as 1.0, make the model more likely to avoid mentioning particular topics provided in the user messages, while lower values, like **0.2**, make the **model less concerned about preventing those topics**.

Why?

Frequency and presence Penalty

The frequency and presence penalties found in the Chat completions API and Legacy Completions API can be used to reduce the likelihood of sampling repetitive sequences of tokens. They work by directly modifying the logits (un-normalized log-probabilities) with an additive contribution.

```
mu[j] -> mu[j] - c[j] * alpha_frequency - float(c[j] > 0) * alpha_presence
```

Where:

- `mu[j]` is the logits of the j-th token
- `c[j]` is how often that token was sampled prior to the current position
- `float(c[j] > 0)` is 1 if `c[j] > 0` and 0 otherwise
- `alpha_frequency` is the frequency penalty coefficient
- `alpha_presence` is the presence penalty coefficient

https://platform.openai.com/docs/guides/gpt/parameter-details

Implement Assignment1 task1 and task2 step by step

Assignment 1: Prompt Engineering

- Download code: https://github.com/LLM-Course/LLM-course.github.io/tree/main/Assignments/Assignment1

- Dependency:
  pip install retrying, openai, urllib3==1.25.11, tqdm, jsonlines

- You are encouraged to complete one and only one task listed below.
  - Task1: LLMs as a knowledgeable doctor
  - Task2: LLMs for AI feedback
  - Task3: LLMs as a data quality evaluator
  - Task4: Using LLMs to remove sensitive info
  - Task5: Jailbreak
  - Task6: Prompt Engineering for public benchmark
  - Task7: Any other topics

Task1: LLMs as a knowledgeable doctor

An example prompt for multi-choice questions

Prompt:
你是一个药剂师考试能手，每次都考100分，这道题对你来说不在话下，深呼吸，并一步一步思考，并给出正确的答案。回答格式为答案是A、B、C或者D。 【最佳选择题】根据健康中国战略，推进健康中国建设主要遵循的原则不包括哪个选项？.
A: "健康优先",
B: "改创新",
C: "科学发展",
D: "公开透明",

Expected Output:
根据健康中国战略，推进健康中国建设主要遵循的原则不包括选项D: "公开透明"。

Input: given Multi-choice questions

Output: select the correct answer.

Criteria: better accuracy.

Code and Data:

- Data description: 100 questions sampled from 2021 National Pharmacist Professional Qualification Examination real questions
- Code and data: `https://github.com/LLM-Course/LLM-course.github.io/tree/main/Assignments/Assignment1/task1`

Resource: https://github.com/LLM-Course/LLM-course.github.io/tree/main/Assignments/Assignment1/task1

Task2: LLMs for AI feedback

An example prompt for AI feedback

Prompt:
Given a questions and two answers. You are a smart guy and please tell which answer better answers the given question.
Questions: Do you know why turkeys became the official food of thanksgiving?

Answer1: To be honest, I don't know anything about that. I know that I'm meant to know a lot about history and current events, but I haven't been programmed with those particular facts, sorry.

Answer2: I know that the American Indians were the first to celebrate the first thanksgiving, and the first European settlers came to New England in the early 1600s, and it is likely that they celebrated the first thanksgiving in the late 1600s. However, it is also likely that some European settlers on the continent celebrated the first thanksgiving in the mid-1500s. A lot of people think that the main factor in thanksgiving is that the settlers in the new world were thankful for finding a new land, and that turkey was their traditional food. Another factor that has often been thought to be important is the Pilgrims' relationship with the Native Americans, and the feast may be one way of trying to show them respect. I can't tell you the definitive answer, but maybe this will help you figure it out?

Expected Output:
Answer2 better answers the given question.

Input: A question `Q` and two answers `a1` and `a2`.

Output: Which one is chosen and rejected

Criteria: The more correlated to the ground truth, the better

Code and Data:

- Data description: 100 answer pairs sampled from huggingface `https://huggingface.co/dat`
- Code and data: `https://github.com/LLM-io/tree/main/Assignments/Assignment`

Resource: https://github.com/LLM-Course/LLM-course.github.io/tree/main/Assignments/Assignment1/task2