

Tùy chỉnh đầu ra bằng các tham số trong OpenAI API

Mục tiêu

Trong bài học này, chúng ta sẽ:

- Khám phá cách tinh chỉnh đầu ra của mô hình ngôn ngữ bằng cách sử dụng các **tham số**.
- Hiểu rõ vai trò của từng tham số: `temperature`, `top_p`, `frequency_penalty`, `presence_penalty`, v.v.
- Trải nghiệm thực tế bằng cách thử nghiệm các tham số ngay trong mã nguồn hoặc trên Playground của OpenAI.

Tổng quan về Tham số trong Chat Completion

OpenAI API cung cấp một số tham số quan trọng để điều chỉnh cách mô hình tạo ra văn bản. Chúng ta sẽ tập trung vào những tham số **thường dùng nhất**:

Tham số	Mô tả ngắn gọn	Giá trị thường dùng
<code>temperature</code>	Điều khiển độ ngẫu nhiên của kết quả. Cao hơn → sáng tạo hơn.	<code>0</code> (chắc chắn), <code>1</code> (trung lập), <code>2</code> (siêu sáng tạo)
<code>top_p</code>	Sử dụng nucleus sampling , mô hình chỉ xét các token có xác suất tích lũy nằm trong <code>p</code> .	0.1 → chỉ lấy top 10%
<code>frequency_penalty</code>	Phạt nếu từ/ý lặp lại nhiều lần .	Từ <code>-2</code> đến <code>2</code>
<code>presence_penalty</code>	Phạt nếu từ/ý đã từng xuất hiện trước đó . Khuyến khích ý tưởng mới.	Từ <code>-2</code> đến <code>2</code>

Ví dụ: Điều chỉnh `temperature`

```
response = openai.ChatCompletion.create(  
    model="gpt-4",  
    messages=[  
        {"role": "user", "content": "Viết một đoạn thơ ngắn về bầu trời đêm"}  
    ],  
    temperature=0.2  
)
```

```
print(response['choices'][0]['message']['content'])
```

“ Với `temperature = 0.2`, kết quả sẽ **logic, nhất quán** và có vẻ "an toàn".

Giờ hãy thử với `temperature = 1.5`

```
response = openai.ChatCompletion.create(
    model="gpt-4",
    messages=[
        {"role": "user", "content": "Viết một đoạn thơ ngắn về bầu trời đêm"}
    ],
    temperature=1.5
)
```

“ Kết quả sẽ **sáng tạo, bất ngờ**, nhưng có thể "quá bay bổng" hoặc không chính xác – gọi là ảo giác(**hallucination**).

Kết hợp các tham số để điều chỉnh hành vi

Hãy xem ví dụ với `presence_penalty` và `frequency_penalty`

```
response = openai.ChatCompletion.create(
    model="gpt-4",
    messages=[
        {"role": "system", "content": "Bạn là một nhà văn sáng tạo."},
        {"role": "user", "content": "Hãy kể một câu chuyện về chú mèo khám phá vũ trụ."}
    ],
    temperature=1.0,
    frequency_penalty=1.2,
    presence_penalty=1.0
)
```

“ Điều này giúp GPT **tránh lặp lại** và **tìm ra nội dung mới mẻ**, tạo nên một câu chuyện hấp dẫn hơn.

Sử dụng Playground để dễ dàng điều chỉnh

Truy cập: <https://platform.openai.com/playground>

- Bạn có thể điều chỉnh tham số qua thanh trượt.
- Playground sẽ **tự động sinh ra mã code Python** để bạn sao chép về dùng.
- Cũng có tính năng **so sánh mô hình** (Compare), giúp bạn đối chiếu đầu ra giữa GPT-4-0125, GPT-4o, hay mô hình fine-tuned của bạn.

Lưu ý khi sử dụng

- **Không nên kết hợp** `top_p` và `temperature` **cùng lúc**, hãy chọn một trong hai.
- Cần **kiểm thử đầu ra nhiều lần**, vì đầu ra không cố định khi `temperature > 0`.
- Nếu GPT tạo ra kết quả kỳ lạ (hallucination), hãy **giảm temperature** và thử lại.

Việc điều chỉnh các tham số trong OpenAI API giúp bạn:

- Tạo ra nội dung **sáng tạo** hơn hoặc **ổn định** hơn tùy theo nhu cầu.
- Tránh việc GPT **lặp lại ý tưởng**.
- Tùy biến trải nghiệm người dùng cho chatbot, hệ thống gợi ý, hoặc sáng tạo nội dung.

Tác giả: Đỗ Ngọc Tú
Công Ty Phần Mềm VHTSoft

Phiên bản #1

Được tạo 5 tháng 5 2025 05:24:34 bởi Đỗ Ngọc Tú

Được cập nhật 6 tháng 5 2025 09:52:34 bởi Đỗ Ngọc Tú