TẬP ĐOÀN BƯU CHÍNH VIỄN THÔNG VIỆT NAM

CÔNG TY CÔNG NGHỆ THÔNG TIN VNPT

LLM API Compatible

TÀI LIỆU MÔ TẢ TÍCH HỢP LLM API

[Thông tin bổ sung]

1. GIỚI THIỆU.

1.1. Mục đích.

Tài liệu mô tả tích hợp API LLM/Embedding Track 2.

1.2. Phạm vi.

Tài liệu bao quát các vấn đề liên quan đến mô tả tích hợp được thiết kế dựa trên tài liệu SRS.

1.3. Đối tượng sử dụng.

Các thí sinh dự thi, tham gia vào cuộc thi VNPT AI - Age of AInicorns - Track 2 - The Builder.

2. QUY TRÌNH.

Các đội thi đăng nhập tại portal của VNPT AI - Age of AInicorns , sau đó có thể Download được API key tại tab Instruction.

3. GIAO TIẾP API.

3.1. [1] Sinh câu trả lời từ LLM Small.

Endpoint : /data-service/v1/chat/completions/vnptai-hackathon-small

Chức năng: Thực hiện hoàn thành hội thoại trò chuyện giữa người dùng và LLM.

Quota: Giới hạn 1000 req/ngày, 60 req/h.

Method: POST

Role:

Request

	Tên trường	Giá trị	Bắt buộc	Mô tả
HEADER	Content-Type	application/json	x	
	Authorization	Bearer ${access_token }	x	Author dịch vụ được cung cấp
	Token-id	String	x	Key dịch vụ được cấp
	Token-key	String	x	Key dịch vụ được cấp

	Tên trường	Kiểu dữ liệu	Mô tả
BODY	***model***	String	Tên mô hình mà BTC cung cấp cho từng tác vụ. Trong API này giá trị là: ***vnptai_hackathon_small***
	messages	List[Dict]	Mảng các đối tượng tin nhắn đại diện cho lịch sử hội thoại. Cấu trúc bao gồm role (system, user, assistant) và content.
	temperature	float	Kiểm soát độ ngẫu nhiên của phân phối xác suất đầu ra (sampling temperature). top_p - number - Optional (Mặc định: 1.0)
	top_p	float	Tham số yêu cầu mô hình chọn nhóm từ sao cho tổng xác suất của chúng đạt đến ngưỡng P.
	top_k	int	Tham số này cầu mô hình chỉ xem xét K từ có xác suất cao nhất.
	n	int	Số lượng câu trả lời được tạo ra cho mỗi input. Tăng giá trị này sẽ nhân chi phí và thời gian xử lý lên n lần.
	stop	String/List	Khi mô hình sinh ra chuỗi này, nó sẽ lập tức

			dừng việc tạo văn bản. Hữu ích để kiểm soát cấu trúc output hoặc ngăn mô hình nói quá dài.
	max_completion_tokens	int	Giới hạn số lượng token tối đa mà mô hình có thể sinh ra ở đầu ra.
	presence_penalty	float	Giá trị từ -2.0 đến 2.0. Phạt các token dựa trên việc chúng đã xuất hiện trong văn bản hay chưa.
	frequency_penalty	float	Giá trị từ -2.0 đến 2.0. Phạt các token dựa trên tần suất xuất hiện của chúng. Giá trị dương làm giảm khả năng mô hình lặp lại nguyên văn cùng một câu.
	response_format	object	Chỉ định định dạng đầu ra. Thiết lập {"type": "json_object"} để đảm bảo mô hình trả về JSON hợp lệ. Cần thiết cho các ứng dụng yêu cầu cấu trúc dữ liệu nghiêm ngặt.
	seed	int	Hỗ trợ tính năng "reproducible outputs". Nếu truyền cùng một seed và các tham số khác không đổi, mô hình sẽ cố gắng trả về kết quả giống hệt nhau (deterministic).
	tools	list	Danh sách các công cụ (functions) mà mô hình có thể gọi. Thay thế cho tham số functions cũ. Cho phép mô hình kết nối với dữ liệu bên ngoài hoặc thực thi hành động.
	tool_choice	string/object	Kiểm soát việc mô hình có bắt buộc phải gọi tool hay không. auto để mô hình tự quyết định, none để ép trả về text, hoặc chỉ định tên hàm cụ thể để ép gọi hàm đó.
	logprobs	boolean	Nếu true, trả về thông tin log probabilities của các token đầu ra. Hữu ích để phân tích độ tự tin

			(confidence) của mô hình đối với câu trả lời.
	top_logprobs	int	Số lượng token có xác suất cao nhất cần trả về tại mỗi vị trí logprobs (0-20).

Ví dụ:

Input	Output

curl --location 'https://api.idg.vnpt.vn/data-service/v1/chat/completions/vnptai-hackathon-small' \
--header 'Authorization: Bearer $AUTHORIZATION' \
--header 'Token-id: $TOKEN_ID' \
--header 'Token-key: $TOKEN_KEY' \
--header 'Content-Type: application/json' \
--data '{
"model": "vnptai_hackathon_small",
"messages": [
{
"role": "user",
"content": "Chào bạn!"
}
],
"temperature": 1.0,
"top_p": 1.0,
"top_k": 20,
"n": 2,
"max_completion_tokens": 512
}'
	{'id': 'chatcmpl-2f2c048b7af24514b3cd6c4cfa0ec97d',
	'object': 'chat.completion',
	'created': 1764754595,
	'model': 'vnptai_hackathon_large',
	'choices': [{'index': 0,
	'message': {'role': 'assistant',
	'content': 'Chào bạn! Tôi là VNPT AI.',
	'refusal': None,
	'annotations': None,
	'audio': None,
	'function_call': None,
	'tool_calls': [],
	'reasoning_content': None},
	'logprobs': None,
	'finish_reason': 'stop',
	'stop_reason': None,
	'token_ids': None},
	{'index': 1,
	'message': {'role': 'assistant',
	'content': 'VNPT AI chào bạn.',
	'refusal': None,
	'annotations': None,
	'audio': None,
	'function_call': None,
	'tool_calls': [],

	'reasoning_content': None}, 'logprobs': None, 'finish_reason': 'stop', 'stop_reason': None, 'token_ids': None}], 'service_tier': None, 'system_fingerprint': None, 'usage': {'prompt_tokens': None, 'total_tokens': None, 'completion_tokens': None, 'prompt_tokens_details': None}, 'prompt_logprobs': None, 'prompt_token_ids': None, 'kv_transfer_params': None}

Giao tiếp bằng mã nguồn Python:

```python
import requests
headers = {
    'Authorization': 'Bearer #Authorization',
    'Token-id': '#TokenID',
    'Token-key': '#TokenKey',
    'Content-Type': 'application/json',
}

json_data = {
    'model': 'vnptai_hackathon_small',
    'messages': [
        {
            'role': 'user',
            'content': 'Hi, VNPT AI.',
        },
    ],
    'temperature': 1.0,
    'top_p': 1.0,
    'top_k': 20,
    'n': 1,
    'max_completion_tokens': 10,
}
```

```
response =
requests.post('https://api.idg.vnpt.vn/data-service/v1/chat/completions/
vnptai-hackathon-small', headers=headers, json=json_data)
response.json()
```

3.2. [2] Sinh câu trả lời từ LLM Large.

Endpoint : /data-service/v1/chat/completions/vnptai-hackathon-large

Chức năng: Thực hiện hoàn thành hội thoại trò chuyện giữa người dùng và LLM.

Quota: Giới hạn 500 req/ngày, 40 req/h.

Method: POST

Role:

Request

	Tên trường	Giá trị	Bắt buộc	Mô tả
HEADER	Content-Type	application/json	x	
	Authorization	Bearer ${access_token }	x	Author dịch vụ được cung cấp
	Token-id	String	x	Key dịch vụ được cấp
	Token-key	String	x	Key dịch vụ được cấp

	Tên trường	Kiểu dữ liệu	Mô tả
BODY	***model***	String	Tên mô hình mà BTC cung cấp cho từng tác vụ.

			Trong API này giá trị là: ***vnptai_hackathon_large***
	messages	List[Dict]	Mảng các đối tượng tin nhắn đại diện cho lịch sử hội thoại. Cấu trúc bao gồm role (system, user, assistant) và content.
	temperature	float	Kiểm soát độ ngẫu nhiên của phân phối xác suất đầu ra (sampling temperature). top_p - number - Optional (Mặc định: 1.0)
	top_p	float	Tham số yêu cầu mô hình chọn nhóm từ sao cho tổng xác suất của chúng đạt đến ngưỡng P.
	top_k	int	Tham số này cầu mô hình chỉ xem xét K từ có xác suất cao nhất.
	n	int	Số lượng câu trả lời được tạo ra cho mỗi input. Tăng giá trị này sẽ nhân chi phí và thời gian xử lý lên n lần.
	stop	String/List	Khi mô hình sinh ra chuỗi này, nó sẽ lập tức dừng việc tạo văn bản. Hữu ích để kiểm soát cấu trúc output hoặc ngăn mô hình nói quá dài.
	max_completion_tokens	int	Giới hạn số lượng token tối đa mà mô hình có thể sinh ra ở đầu ra.
	presence_penalty	float	Giá trị từ -2.0 đến 2.0. Phạt các token dựa trên việc chúng đã xuất hiện trong văn bản hay chưa (bất kể tần suất)
	frequency_penalty	float	Giá trị từ -2.0 đến 2.0. Phạt các token dựa trên tần suất xuất hiện của chúng. Giá trị dương làm giảm khả năng mô hình lặp lại

			nguyên văn cùng một câu.
	response_format	object	Chỉ định định dạng đầu ra. Thiết lập {"type": "json_object"} để đảm bảo mô hình trả về JSON hợp lệ. Cần thiết cho các ứng dụng yêu cầu cấu trúc dữ liệu nghiêm ngặt.
	seed	int	Hỗ trợ tính năng "reproducible outputs". Nếu truyền cùng một seed và các tham số khác không đổi, mô hình sẽ cố gắng trả về kết quả giống hệt nhau (deterministic).
	tools	list	Danh sách các công cụ (functions) mà mô hình có thể gọi. Thay thế cho tham số functions cũ. Cho phép mô hình kết nối với dữ liệu bên ngoài hoặc thực thi hành động.
	tool_choice	string/object	Kiểm soát việc mô hình có bắt buộc phải gọi tool hay không. auto để mô hình tự quyết định, none để ép trả về text, hoặc chỉ định tên hàm cụ thể để ép gọi hàm đó.
	logprobs	boolean	Nếu true, trả về thông tin log probabilities của các token đầu ra. Hữu ích để phân tích độ tự tin (confidence) của mô hình đối với câu trả lời.
	top_logprobs	int	Số lượng token có xác suất cao nhất cần trả về tại mỗi vị trí logprobs (0-20).

Ví dụ:

Input	Output
`curl --location 'https://api.idg.vnpt.vn/data-service/v1/`	`{'id': 'chatcmpl-2f2c048b7af24514b3cd6c4cfa0ec97d`

```
chat/completions/vnptai-hackathon-large'
\
--header 'Authorization: Bearer
$AUTHORIZATION' \
--header 'Token-id: $TOKEN_ID' \
--header 'Token-key: $TOKEN_KEY' \
--header 'Content-Type: application/json'
\
--data '{
    "model": "vnptai_hackathon_large",
    "messages": [
        {
            "role": "user",
            "content": "Chào bạn!"
        }
    ],
    "temperature": 1.0,
    "top_p": 1.0,
    "top_k": 20,
    "n": 2,
    "max_completion_tokens": 512
}'
```

```
',
 'object': 'chat.completion',
 'created': 1764754595,
 'model': 'vnptai_hackathon_large',
 'choices': [{'index': 0,
   'message': {'role': 'assistant',
    'content': 'Chào bạn! Tôi là VNPT
AI.',
    'refusal': None,
    'annotations': None,
    'audio': None,
    'function_call': None,
    'tool_calls': [],
    'reasoning_content': None},
   'logprobs': None,
   'finish_reason': 'stop',
   'stop_reason': None,
   'token_ids': None},
  {'index': 1,
   'message': {'role': 'assistant',
    'content': 'VNPT AI chào bạn.',
    'refusal': None,
    'annotations': None,
    'audio': None,
    'function_call': None,
    'tool_calls': [],
    'reasoning_content': None},
   'logprobs': None,
   'finish_reason': 'stop',
   'stop_reason': None,
   'token_ids': None}],
 'service_tier': None,
 'system_fingerprint': None,
 'usage': {'prompt_tokens': None,
  'total_tokens': None,
  'completion_tokens': None,
  'prompt_tokens_details': None},
 'prompt_logprobs': None,
```

	'prompt_token_ids': None, 'kv_transfer_params': None}

Giao tiếp bằng mã nguồn Python:

```
import requests
headers = {
    'Authorization': 'Bearer #Authorization',
    'Token-id': '#TokenID',
    'Token-key': '#TokenKey',
    'Content-Type': 'application/json',
}

json_data = {
    'model': 'vnptai_hackathon_large',
    'messages': [
        {
            'role': 'user',
            'content': 'Hi, VNPT AI.',
        },
    ],
    'temperature': 1.0,
    'top_p': 1.0,
    'top_k': 20,
    'n': 1,
    'max_completion_tokens': 10,
}

response =
requests.post('https://api.idg.vnpt.vn/data-service/v1/chat/completions/
vnptai-hackathon-large', headers=headers, json=json_data)
response.json()
```

3.3. [3] Tính toán biểu diễn của đoạn văn bản

Endpoint : /data-service/vnptai-hackathon-embedding

Chức năng: Tính toán biểu diễn của đoạn văn bản.

Quota: 500 req/m.

Method: POST

Role:

Request

	Tên trường	Giá trị	Bắt buộc	Mô tả
HEADER	Content-Type	application/json	x	
	Authorization	Bearer ${access_token}	x	Author dịch vụ được cung cấp
	Token-id	String	x	Key dịch vụ được cấp
	Token-key	String	x	Key dịch vụ được cấp

	Tên trường	Kiểu dữ liệu	Mô tả
BODY	***model***	String	Tên mô hình mà BTC cung cấp cho tác vụ embedding. Trong trường hợp này là: ***vnptai_hackathon_embedding***
	input	String	Văn bản đầu vào cần được vector hoá.
	encoding_format	String	Kiểm soát độ ngẫu nhiên của phân phối xác suất đầu ra (sampling temperature). top_p - number - Optional (Mặc định: 1.0)

Ví dụ:

Input	Output

```
curl --location
'https://api.idg.vnpt.vn/data-service/vnptai-hackathon-embedding' \
--header 'Authorization: Bearer $AUTHORIZATION' \
--header 'Token-id: $TOKEN_ID' \
--header 'Token-key: $TOKEN_KEY' \
--header 'Content-Type: application/json' \
--data '{
    "model": "vnptai_hackathon_embedding",
    "input": "Xin chào VNPT AI",
    "encoding_format": "base64"
}'
```

```
'data': [{'index': 0,
   'embedding': [-0.044116780161857605,
    -0.021570704877376556,
    -0.033462729305028915,
    0.008436021395027637,
    -0.041678354144096375,
    -0.05991028994321823,
    0.010203881189227104,
    0.009467664174735546,
    0.025847330689430237,
    0.0431038960814476,
    0.014639943838119507,
    -0.00491905864328146,
    -0.0030832039192318916,
    0.024440545588731766,
    0.02485320344567299,
    -0.0067572579719126225,
    0.013420729897916317,
    -0.007530989591032267,
    0.008604835718870163,
    -0.045054636895656586,
    -0.03134317323565483,
    0.03627629950642586,
    -0.010119474492967129,
    -0.008440710604190826,
...
'model': 'vnptai_hackathon_embedding',
 'logID':
'06c03a76-d0be-11f0-b581-11d943acf105-6bc48dbe-Zuulserver',
 'id':
'embd-7db936b0881543cea0b95b5c182abd09',
 'object': 'list',
 'challengeCode': '11111'}
```

Giao tiếp bằng mã nguồn Python:

```
import requests
headers = {
   'Authorization': 'Bearer #Authorization',
   'Token-id': '#TokenID',
   'Token-key': '#TokenKey',
   'Content-Type': 'application/json',
}

json_data = {
   'model': 'vnptai_hackathon_embedding',
   'input': 'Xin chào, mình là VNPT AI.',
   'encoding_format': 'float',
}
response =
requests.post('https://api.idg.vnpt.vn/data-service/vnptai-hackathon-emb
edding', headers=headers, json=json_data)
response.json()
```

4. NHỮNG VẤN ĐỀ KHÁC.

<Những yêu cầu đặc thù khác sẽ được bổ sung trong quá trình phát triển hệ thống sau này>

5. TÀI LIỆU THAM KHẢO.

5.1. Tài liệu của dự án:

STT	Mã	Tên tài liệu	Vị trí lưu trữ	Ghi chú

5.2. Các tài liệu khác (sách, báo, tiêu chuẩn, quy chuẩn, luật, nghị định, thông tư, …).

STT	Mã	Tên tài liệu	Vị trí lưu trữ	Ghi chú