

TP - Découverte de GPT avec un notebook dans le cloud

Objectif

Explorer les capacités de GPT via l'API OpenAI pour générer des textes résumés ou créatifs en ajustant les paramètres du modèle. Ce TP utilise un notebook pour offrir une expérience interactive et didactique.

Préparer l'Environnement

1. Utilisation d'un Notebook dans le Cloud

- Les notebooks (par exemple, Jupyter ou Google Colab) permettent d'exécuter du code en blocs interactifs, parfait pour des expérimentations en intelligence artificielle.
- Google Colab est un excellent choix :
 - Gratuit (avec des options payantes pour un GPU/TPU avancé).
 - Compatible avec Python et doté de bibliothèques préinstallées.
 - Accessible depuis un navigateur.

Configuration de Google Colab :

1. Connectez-vous à Google Colab.
2. Créez un nouveau notebook.
3. Ajoutez une cellule de code pour installer les bibliothèques nécessaires.

2. Installer les Bibliothèques

Utilisez la bibliothèque `openai` pour interagir avec le modèle GPT. Pour générer des visualisations, ajoutez également `matplotlib`.

Installer les bibliothèques dans Google Colab :

```
# Installer la bibliothèque OpenAI  
  
!pip install openai
```

3. Comprendre les Paramètres Principaux

- **temperature** : Contrôle la créativité des réponses (valeurs entre 0 et 1).
 - Basse température (ex. : 0.2) → Réponses cohérentes mais conservatrices.
 - Haute température (ex. : 0.8) → Réponses plus créatives mais parfois incohérentes.
- **max_tokens** : Nombre maximal de mots dans la réponse générée.
 - Limite la longueur de la réponse pour éviter des textes trop longs.
- **top_p** : Nucleus sampling, qui ajuste la probabilité cumulée des choix de mots.

- Valeur basse → Génération plus prévisible.
- Valeur élevée → Génération plus diversifiée.

4. Découverte des capacités de GPT

1. Configuration

Un exemple de code pour se connecter à l'API OpenAI :

```
# Importer les bibliothèques nécessaires
import openai

# Configurer la clé API OpenAI
# Remplacez "VOTRE_CLE_API" par votre clé OpenAI
openai.api_key = "VOTRE_CLE_API"

# Fonction pour générer du texte
def generate_text(prompt, temperature=0.7, max_tokens=100, top_p=1.0):
    response = openai.Completion.create(
        engine="text-davinci-003", # Utiliser GPT-3.5 ou supérieur
        prompt=prompt,
        temperature=temperature,
        max_tokens=max_tokens,
        top_p=top_p
    )
    return response.choices[0].text.strip()

# Tester la connexion avec une question simple
prompt = "Expliquez en une phrase pourquoi les chats ronronnent."
response = generate_text(prompt)
print("Réponse générée :\n", response)
```

2. Tâches

1. Générer un résumé d'un texte donné :

- Exemple de texte :
"L'intelligence artificielle (IA) est un domaine de l'informatique qui se concentre sur la création de systèmes capables de réaliser des tâches nécessitant habituellement une intelligence humaine, telles que la reconnaissance vocale, la vision par ordinateur et la prise de décision."
- Fournissez un prompt pour résumer ce texte :

```

passage = """
L'intelligence artificielle (IA) est un domaine de l'informatique qui se con
capables de réaliser des tâches nécessitant habituellement une intelligence
la vision par ordinateur et la prise de décision.
"""

prompt = f"Résumé ce texte en 50 mots ou moins :\n{passage}"
summary = generate_text(prompt, temperature=0.7, max_tokens=50)
print("Résumé généré :\n", summary)

```

Modifier les paramètres pour observer les différences :

- Essayez deux valeurs de `temperature` (0.2 et 0.8).
- Analysez l'impact sur la créativité et la précision du résumé :

```

low_temp = generate_text(prompt, temperature=0.2, max_tokens=50)
high_temp = generate_text(prompt, temperature=0.8, max_tokens=50)

print("Résumé avec température basse (0.2) :\n", low_temp)
print("\nRésumé avec température élevée (0.8) :\n", high_temp)

```

Exploration créative

- **Tâche : Générer une histoire courte avec une contrainte.**
 - Exemple : "Une histoire avec un chat, une horloge et une tempête."
 - Prompt :

```

creative_prompt = "Raconte une histoire impliquant un chat, une horloge cassée
story = generate_text(creative_prompt, temperature=0.9, max_tokens=100)
print("Histoire générée :\n", story)

```

En résumé:

Analyse et Discussion

1. Impact des paramètres :

- Identifiez les variations dans les réponses selon les paramètres.
- Exemple : avec une température élevée, le modèle peut introduire des éléments inattendus ou incohérents.

2. Forces et limites :

- **Forces** : Créativité, fluidité dans le texte.
- **Limites** : Cohérence parfois fragile, biais ou contenu non pertinent.

3. **Applications possibles :**

- Résumé automatique, rédaction assistée, création de contenu.