Rendu 3 : SAE IA

Consigne :

Prise en main de GPT-2 et faire des tests pour interroger le modèle pré entraîné avec Jupyter Notebook et Tensorflow.

GPT-2 est un modèle IA spécialisé pour le langage développé par OpenAI. Il a été entraîné sur une base de plus de 7,000 livres de fictions et de multiples pages web.

Début de la prise en main :

Nous avons chargé le modèle pré-entraîné à l'aide du module transformers qui traite des modèles séquences à séquence. L'avantage est que le modèle va générer un résultat plus pertinent car il effectue à traitement en mettant en relation chaque mot.

```
model = TFGPT2LMHeadModel.from_pretrained("gpt2”)
```

```
tokenizer = GPT2Tokenizer.from_pretrained("gpt2", padding_side="left")
```

Le tokenizer permet de traiter les données pour les envoyer correctement en entrée du modèle. Le padding permet d'uniformiser de manière fixe la taille des différentes parties du texte tronqué.

Code :

```
!pip install transformers

# Importer les librairies pour interroger le modèle pré-entraîné
# GPT2Tokenizer pour tokenizer les inputs
# TFGPT2LMHeadModel pour interroger le model pré-entrainé

from transformers import TFGPT2LMHeadModel, GPT2Tokenizer

# Charger le tokenizer et le modèle pré-entraîné

tokenizer = GPT2Tokenizer.from_pretrained("gpt2", padding_side="left")
model = TFGPT2LMHeadModel.from_pretrained("gpt2", from_pt=False,
pad_token_id=tokenizer.eos_token_id)

# Texte d'entrée, à compléter par le modèle pré-entraîné
```

```
input_text = "Try to"


# Tokenizer le texte d'entrée
input_ids = tokenizer.encode(input_text, return_tensors='tf')

# Interroger le modèle pré-entraîné avec le texte d'entrée tokenizé

output = model.generate(input_ids, max_length=100,do_sample=True,
no_repeat_ngram_size=3,temperature=0.8)

# Afficher le texte généré par le modèle pré-entraîné

print("Output:\n" + 100 * '-')
print(tokenizer.decode(output[0], skip_special_tokens=True))
```

Le paramètre `no_repeat_ngram_size=3` permet d'éviter les répétitions non logiques du modèle lors de la génération du texte. De plus avec le paramètre température, on inclut les données ayant au moins 80% de pertinence. Celà permet d'obtenir des réponses variées et pertinentes comme :

Output:

Try to see where you go wrong

It is not about making sure you are safe. You need to be able to avoid being killed by other people and you want to keep your friends safe. If your friends are going to harm you or you are going after your own friends (especially a former lover) it is only right that you stay safe and keep your family safe.

Do not be afraid to talk about it. Even the people you are thinking about might be thinking about you now

Lors du décodage, nous utilisons le paramètre `skip_special_tokens=True` pour permettre de ne pas traiter les tokens utilisés pour le padding.

Conclusion :

Grâce au modèle pré-entraîné GPT-2 sous forme de transformer, nous avons directement pu interroger celui-ci à l'aide de différents textes qu'il a pu compléter avec pertinence suivant sa base de connaissance. Ce travail nous a permis de comprendre le fonctionnement de ce modèle, pour ainsi aisément aborder l'entraînement et le fine-tuning ensuite.