

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-entraîné

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 tokenisé

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))

```

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

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. Cela 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

Avec des prompts un peu plus complet et technique :

```
input_text = "What a nuclear submarine is made of ?"
```

Output:

What a nuclear submarine is made of?

I think it is the most advanced submarine in the world. It has a huge range so it can operate at up to 150 km per hour. The main problem is that it only has 5 tons of fuel.

The submarine is the biggest weapon in the entire world but if we had the atomic bomb, it would be one of the biggest weapons in the history of mankind.

```
input_text = "What is the best way to learn a language ?"
```

Output:

What is the best way to learn a language? The best way is to learn by reading. And read by talking. And learn by interacting with people."

The next language is the one that most people talk about.

It's not about being successful, it's "doing what you love," and the best answer is "doing more than what you need to do."

,

– The English Language Project

"There are so many great languages out there, and they

Nous pouvons donc voir que certaines réponses sont tronquées du au max lenght mais que le modèle n'est pas assez affuté pour véritablement comprendre ce qui est dit et de cerner l'enjeu et ne fait que compléter le texte.

C'est la limite de ce modèle. C'est pourquoi le fine-tuning avec l'aide de notre dataset sera requis.

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.