Equipe cancer - LIVRABLE 2.2

Prise en main de GPT-2

Compréhension de GPT-2 :

GPT-2, développé par OpenAI, est un modèle de traitement du langage naturel (NLP) qui a été introduit en 2019. GPT-2 est principalement utilisé pour générer automatiquement du texte.

Architecture du modèle : GPT-2 est un modèle de traitement du langage naturel basé sur l'architecture Transformer. Il utilise une série de couches Transformer, composées d'encodeurs et de décodeurs. Les encodeurs analysent les relations entre les mots dans une séquence, tandis que les décodeurs génèrent du texte de manière auto-régressive. GPT-2 utilise des encodages positionnels pour tenir compte de l'ordre des mots. Il peut être adapté à diverses tâches spécifiques grâce au fine-tuning.

Prise en main de GPT-2:

Chargement du modèle

GPT 2 TOKENIZER

https://keras.io/api/keras_nlp/models/gpt2/gpt2_tokenizer/

Permet de transformer une phrase en un tableau d'indice en fonction du vocabulaire que nous avons choisis car nous ne pouvons pas mettre du texte en entrée du modèle

```
[ ]  tokenizer = keras_nlp.models.GPT2Tokenizer.from_preset("gpt2_base_en") # presets gpt2_medium_en gpt2_large_en
```

```
Downloading data from https://storage.googleapis.com/keras-nlp/models/gpt2_base_en/v1/vocab.json
1042301/1042301 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/keras-nlp/models/gpt2_base_en/v1/merges.txt
456318/456318 [==============================] - 0s 0us/step
```

```
[ ]  print(tokenizer("i am a super car "))

     tokenizer.detokenize(tokenizer("i am a super car"))
```

```
tf.Tensor([  72  716  257 2208 1097  220], shape=(6,), dtype=int32)
<tf.Tensor: shape=(), dtype=string, numpy=b'i am a super car'>
```

Interrogation de GPT-2

Afin d'effectuer nos tests nous avons utilisé un modèle pré-entraîné de GPT2 et généré la suite de phrase.

Nous n'avons pas encore trouvé le moyen de lui donner un contexte de question/réponse, cependant, nous avons remarqué que si un point d'interrogation ou un double point était présent à la fin de la phrase de départ le modèle l'interprétant comme un dialogue question/réponse.

Nous avons donc décidé de faire une fonction QA permettant simplement de retourner la réponse sans la question au début.

```
[ ] print(gpt2_lm.generate('Give me a list of different types of cancer.', max_length=60))

    Give me a list of different types of cancer.

    1) Cancer with a name.

    I'm sure there are a few people out there who think this is a bad idea. But it's a great idea. It can save lives. It's a way of getting a little
```

Exemple du type de génération que nous avons avec l'input au début et un début de réponse

GPT2 en réponse à des stimuli

```
[3] !pip install keras_nlp
```

```
import numpy as np
from keras_nlp.models import GPT2CausalLM, GPT2CausalLMPreprocessor

gpt2_lm = GPT2CausalLM.from_preset("gpt2_base_en")
def QA(input,max_length=60):
  return gpt2_lm.generate(input, max_length)[len(input):]
print(QA('You are a cancer dectector robot and i want to know if i have a thyroid cancer?',100))
```

```
Using TensorFlow backend


You can help by sharing your story with the world.

I have thyroid cancer, I have an infection, and i have been treated for it. What is it?

I have been treated for thyroid cancer since January 2011. I am now on a waiting list. How does it work?

You have to register for your cancer diagnosis. If you don't,
```

Exemple de la génération sans l'input.
QA prend les mêmes paramètres que la fonction generate : input et max_length : nombre de caractères

Tests et évaluation de GPT-2

3.1 Choix des scénarios de test

La sélection des scénarios de test constitue une étape cruciale dans l'évaluation de la compréhension de chatGPT. Dans le cadre de cette étude, nous avons opté pour des simulations de situations de questions/réponses (Q&R) portant sur notre base de données dédiée aux cancers. Cette section vise à détailler les raisons motivant le choix de ces scénarios particuliers et à offrir un aperçu des situations de test évaluées.

Les Situations de Test Évaluées :

Les situations de test ont été spécifiquement conçues pour évaluer la capacité de GPT2 à comprendre et à répondre de manière précise à un large éventail de questions liées aux cancers.

Pourquoi Ces Scénarios de Test Ont Été Sélectionnés :

La sélection des scénarios de test a été orientée par plusieurs objectifs. Tout d'abord, nous avons cherché à évaluer la capacité de ChatGPT à traiter de manière précise des informations médicales spécialisées, en se concentrant sur le domaine complexe des cancers. Ensuite, nous avons choisi des scénarios qui abordent diverses questions, allant des requêtes générales aux demandes plus spécifiques, dans le but de mesurer la polyvalence du modèle.

En résumé, la sélection des scénarios de test vise à améliorer la capacité du modèle à comprendre et à répondre de manière robuste à une variété de questions médicales complexes, reflétant la diversité des interrogations potentielles rencontrées dans un contexte réel de communication sur les cancers.

3.2 Méthodologie de test

Dans cette section, nous détaillerons la mise en place des tests, en exposant le processus déployé pour évaluer les performances de chatGPT dans le cadre des scénarios de question/réponse sur les cancers.

Mise en Place des Tests :

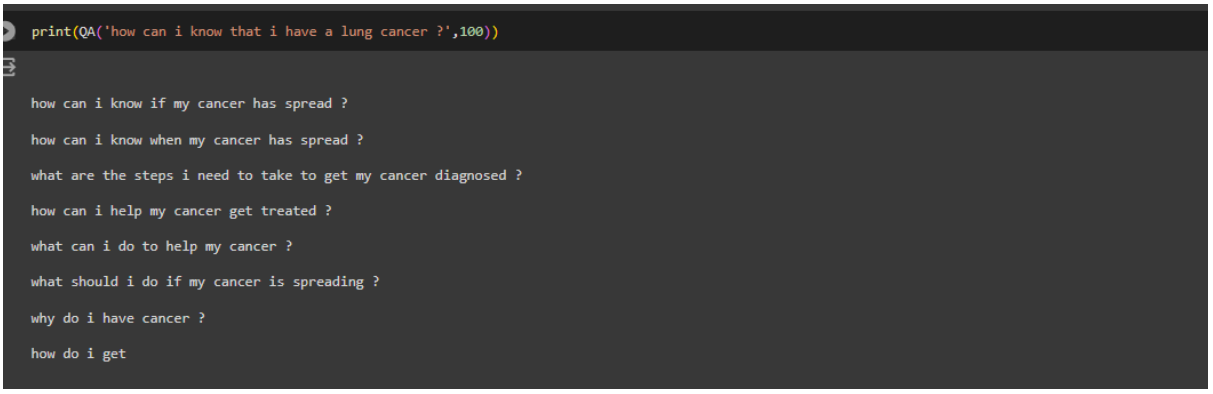
Les tests ont été conçus pour poser des questions types qu'un utilisateur pourrait transmettre au modèle. C'est-à-dire en anglais en utilisant le terme “cancer” dans nos phrases test. L'objectif étant d'identifier les réponses les plus pertinentes.

3.3 Résultats des tests

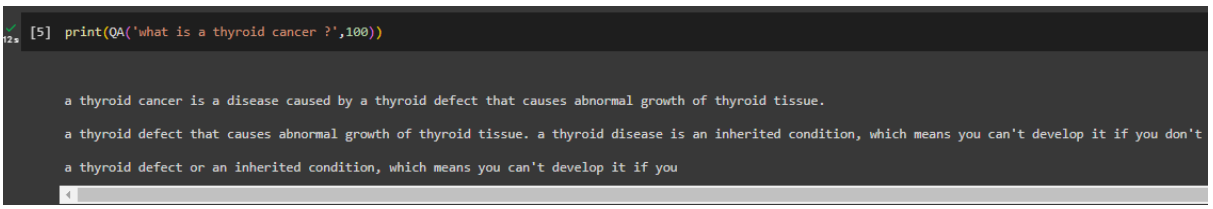
Cette partie se focalise sur la présentation des résultats issus des tests effectués. Elle englobe un résumé des performances de ChatGPT dans chaque scénario évalué, accompagné d'une analyse des résultats. Cela vise à fournir une compréhension des limitations du modèle.

Analyse des Résultats :

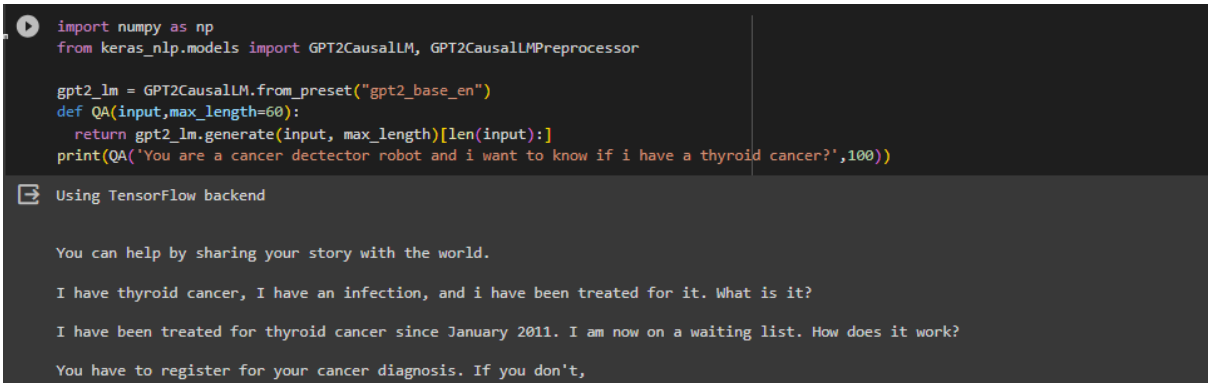
Le modèle a des limites car il n'est pas encore très précis dans ces réponses, il fait encore certaines erreurs. Cette analyse servira de base à la discussion sur la capacité du modèle à comprendre et à répondre efficacement aux questions médicales complexes.



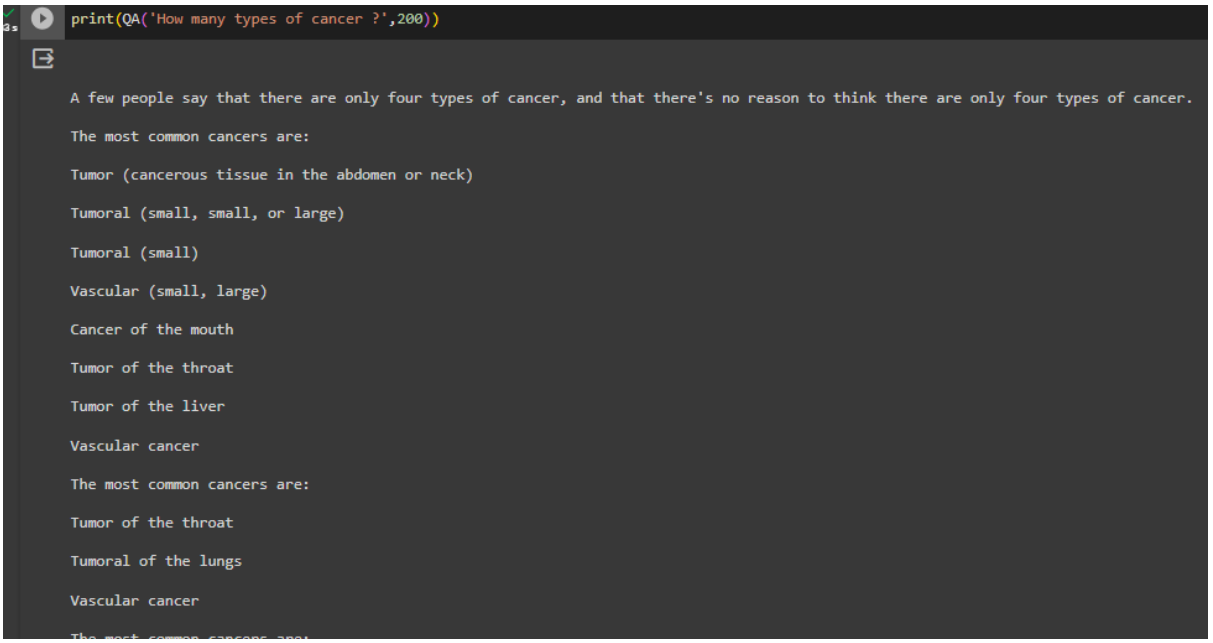
Dans cet exemple, on peut observer que l'on demande au modèle comment savoir si on a un cancer du poumon mais celui ne comprend pas et répond par une série de questions.



Dans cet exemple, on demande la définition d'un cancer de la thyroïde. Le modèle commence prometteuse en fournissant une définition concise en une phrase, et va même plus loin en définissant un des termes inclus dans cette phrase, à savoir la dysgénésie thyroïdienne (thyroid defect).



Dans cet exemple, on informe GPT-2 qu'il est un robot détecteur de cancer, et on lui pose la question de savoir si l'on a un cancer de la thyroïde. Il semble ne pas bien comprendre la demande, car il nous demande un témoignage, mais ensuite, il poursuit de lui-même en fournissant un témoignage de toute pièce.



Dans cet exemple, on demande à GPT2, les différents types de cancer et le résultat qu'il donne est plutôt cohérent malgré le fait qu'il y ait plusieurs répétitions.On observe également la présence du cancer de la poitrine (poumon), l'un des types de cancer inclus dans le jeu de données Kaggle que nous utiliserons par la suite.

Conclusion

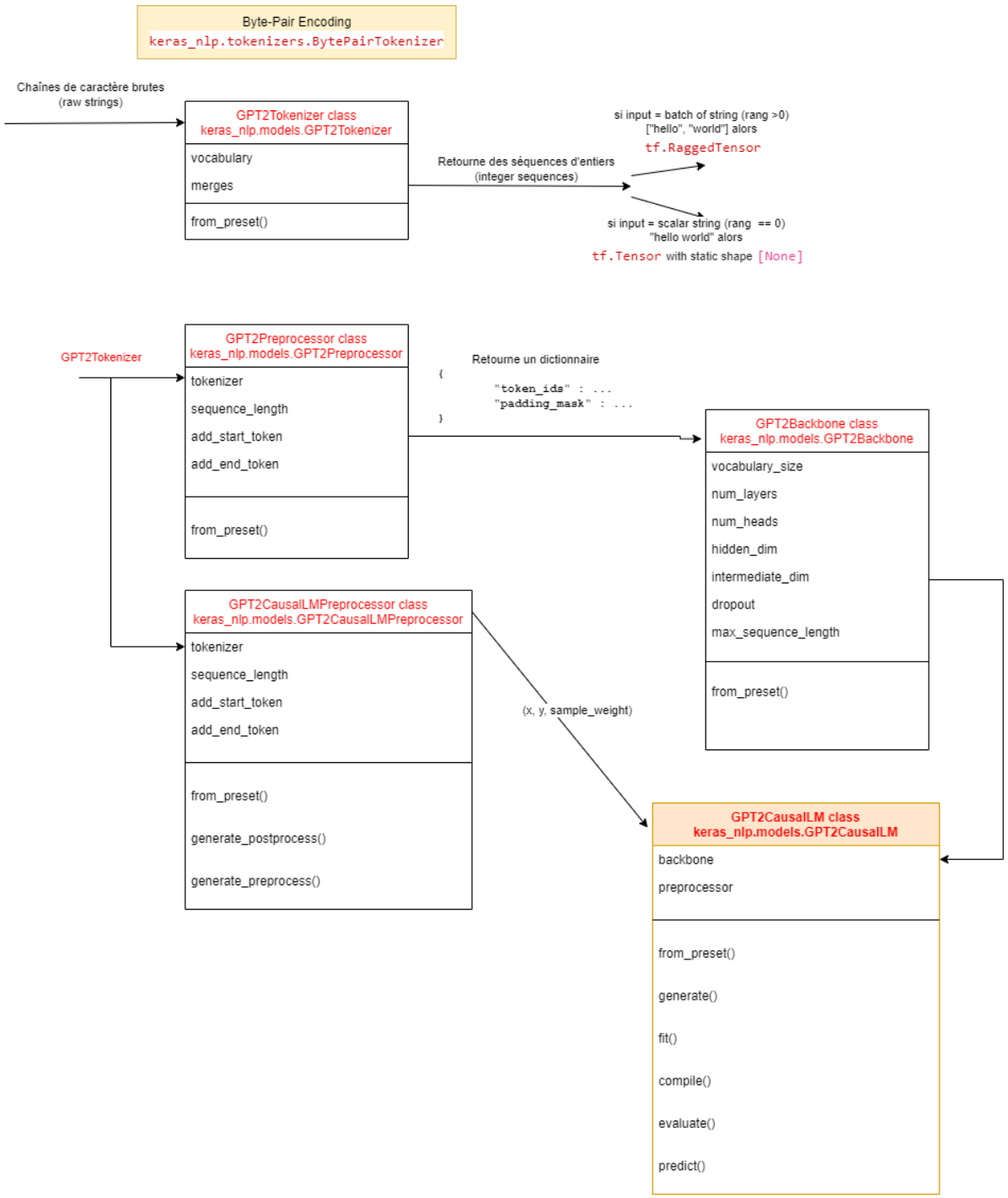
Bien que le modèle puisse nous fournir des réponses concernant les types de cancer, nous pouvons également constater qu'il nous donne très souvent des réponses aléatoires ou des réponses peu précises.

Les résultats peu concluants sont tout à fait attendus car nous avons utilisé dans nos générations un preset du modèle et non pas un modèle personnalisé.
Nous sommes optimistes quant à l'amélioration des résultats après avoir affiné le modèle avec notre propre ensemble de données.

Nous discuterons des développements futurs pour notre projet, en identifiant les pistes d'amélioration et les ajustements potentiels nécessaires pour optimiser l'utilisation de chatGPT dans le domaine médical.

Annexe:

Lien du notebook Google Colab de test : Compréhension de GPT2.ipynb


Byte-Pair Encoding
keras_nlp.tokenizers.BytePairTokenizer
Chaînes de caractère brutes
(raw strings)
GPT2Tokenizer class
keras_nlp.models.GPT2Tokenizer
vocabulary
merges
from_preset()
Retourne des séquences d'entiers
(integer sequences)
si input = batch of string (rang >0)
["hello", "world"] alors
tf.RaggedTensor
si input = scalar string (rang == 0)
"hello world" alors
tf.Tensor with static shape [None]
GPT2Tokenizer
GPT2Preprocessor class
keras_nlp.models.GPT2Preprocessor
tokenizer
sequence_length
add_start_token
add_end_token
from_preset()
Retourne un dictionnaire
{
"token_ids" : ...
"padding_mask" : ...
}
GPT2Backbone class
keras_nlp.models.GPT2Backbone
vocabulary_size
num_layers
num_heads
hidden_dim
intermediate_dim
dropout
max_sequence_length
from_preset()
GPT2CausalLMPreprocessor class
keras_nlp.models.GPT2CausalLMPreprocessor
tokenizer
sequence_length
add_start_token
add_end_token
from_preset()
generate_postprocess()
generate_preprocess()
(x, y, sample_weight)
GPT2CausalLM class
keras_nlp.models.GPT2CausalLM
backbone
preprocessor
from_preset()
generate()
fit()
compile()
evaluate()
predict()

Glossaire :

Transformer : Une architecture de réseau de neurones utilisée dans GPT-2, qui repose sur l'attention multi-tête pour modéliser les relations entre les mots dans une séquence.

TensorFlow/Keras : Des bibliothèques de machine learning largement utilisées pour la construction et l'entraînement de modèles de réseaux de neurones, dont GPT-2.

Pré-entraînement : Le processus d'entraînement initial d'un modèle sur un grand corpus de données avant de l'adapter à des tâches spécifiques.

Modèle de langage : Un modèle statistique ou un réseau de neurones utilisé pour prédire la probabilité des mots ou des caractères suivants dans une séquence de texte.

Encodeur : Une partie d'une couche Transformer qui prend en entrée la séquence de mots et effectue des opérations pour représenter l'information de manière hiérarchique.

Décodeur : La partie d'une couche Transformer responsable de la génération de texte, utilisant la représentation de la séquence pour prédire le mot suivant.

Auto-régressif : Un modèle qui génère du texte en prédisant un mot à la fois, en utilisant les mots précédents comme contexte.

Fine-Tuning : Le processus d'adapter un modèle pré-entraîné à des tâches spécifiques en utilisant des données d'entraînement supplémentaires.

Encodage positionnel : Une technique utilisée dans les modèles Transformer pour tenir compte de l'ordre des mots dans une séquence.