Test technique

Tu trouveras ici le test technique de Xerfi pour le poste de développeur Python. Ce test comporte deux exercices :

- Un premier très orienté sur l'algorithmie.
- Un deuxième plus orienté sur ce que tu pourrais être amené à faire comme mission chez Xerfi.

L'objectif de ces exercices est d'évaluer ta façon de coder et ta logique. Ces deux points importent plus que le résultat de l'algorithme. Concernant le premier exercice, il existe une correction sur internet. A toi de voir si tu souhaites la consulter ou non. Ces deux exercices sont relativement courts, je ne souhaite pas que tu y passes des dizaines d'heures. Si tu te sens bloqué ou s'il y a un point que tu ne comprends pas, n'hésites pas à me contacter pour me poser des questions.

Il y a trois contraintes à respecter :

- Les codes doivent être réalisés en Python
- Les codes doivent être déposés sur github et je dois y avoir accès (rlorenzi@xerfi.fr)
- Je dois pouvoir tout lancer sans modifier la moindre ligne

Concernant le temps que tu as pour réaliser ces exercices, il sera mentionné par email.

Bonne chance !

1er exercice : le pipotron

Un pipotron est un générateur automatique de phrases. Le principe est que la phrase est conçue en ajoutant plusieurs composants les uns derrière les autres, chaque composant étant tiré au hasard. Par exemple, pour un pipotron à 2 composants, on tire une première possibilité parmi un ensemble de mots (il ne s'agit pas forcément d'un mot unique, il faut plus voir ça comme un bout de phrase), puis on réitère l'opération sur un deuxième ensemble et on met les deux tirages bout à bout. A la fin, cela nous donne une phrase pleine de sens (en théorie).

L'objectif de cet exercice est de déterminer si une phrase donnée est un pipotron ou non. Pour cela, l'algorithme prendra en entrée les paramètres suivants :

- **p_size** : un entier supérieur ou égal à 2 représentant la taille du pipotron
- **list_possibilities** : un dictionnaire contenant l'ensemble des possibilités pour chaque composant du pipotron (voir exemple ci-dessous). Attention ! Le dernier composant du pipotron sera toujours une ponctuation !
- **sentence** : une phrase au format string qui correspond à la phrase à tester

Vous pouvez utiliser les librairies que vous voulez, les méthodes que vous voulez. La seule contrainte est que ce code soit fait en Python.

Exemple d'un test possible

On considère les paramètres suivants :

p_size = 3
dict_possibilities = {1 : ['Je', 'Tu', 'Demain je'],
2 : ['mange', 'bois', 'vole', 'voles'],
3 : ['.', ' !', ' ?', '...']}
sentence_1 : 'Je mange !'
sentence_2 : 'Tu dors ?'

Le résultat attendu sera alors :

my_function(p_size, dict_possibilities, sentence_1) = True
my_function(p_size, dict_possibilities, sentence_2) = False

2ème exercice : plongement lexical

En passant par langchain, peux-tu intégrer dans une base ChromaDB ce document word ?
De plus je veux que tu modifies ses metadata :

- Le numéro de la page doit être multiplié par 3.
- Je veux que tu ajoutes la date du jour
- La source doit être : test_technique_xerfi.pdf

Enfin, je souhaite également une fonction qui me permette d'ajouter du texte au début du contenu d'un document précédemment ingérer dans la base Chroma, puis de le mettre à jour dans la base avec le nouveau contenu. Ce document est récupéré grâce à son id.

Plusieurs choses sont à prendre en compte pour cet exercice.

- La base ChromaDB est juste à mettre en local, il ne faut pas la déployer.
- Tout doit se faire avec des méthodes gratuites. Si jamais tu vois que certaines choses nécessitent de passer par des API payantes, tu me dis et on avise. Mais en théorie, tout peut se faire en passant par des solutions gratuites.

Pour résumer, il y a 4 choses à faire dans cet exercice :

1. Créer la base Chroma en local
2. Y ajouter ce document en prenant soin de séparer les pages
3. Modifier ses metadata
4. Faire une fonction permettant de modifier un document existant dans la base Chroma.

Petite précision pour le point 4 :
La fonction doit avoir cette tête-là :

update_doc(text_to_add, doc_id)

Si le contenu du doc est initialement le suivant : « Il était une fois une histoire passionnante blablabla … » et qu'on souhaite ajouter le texte « Bien le bonjour ! », alors le contenu du document après sa modification sera « Bien le bonjour ! Il était une fois une histoire passionnante blablabla … »
Les metadata de ce document ne doivent pas être modifiées au cours du changement.