TP 17 : Perfectionner un chat bot +

Améliorer l'Interface Homme/Machine

Bienvenue dans ce module qui va te permettre d'enrichir la compréhension de ton chabot…

L'objectif est de permettre à ton programme de comprendre des phrases similaires (comme "je veux faire une recherche internet" ou "je veux chercher une vidéo"). On peut utiliser un système de **reconnaissance d'intentions avancé**. Cela implique d'analyser les mots-clés dans la phrase et d'identifier leur intention principale grâce à des techniques simples de traitement du langage naturel (NLP).
Voici une solution détaillée :

Étapes pour comprendre des phrases similaires

1. Ajouter une étape d'analyse de mots-clés

Ajoutons un module pour détecter les intentions à partir de mots-clés. L'idée est de comparer les mots présents dans la phrase de l'utilisateur avec ceux qui sont associés à chaque intention.

2. Exemple de base d'intentions enrichie

Dans le fichier `base_de_donnees.json`, tu peux définir des intentions avec des listes de mots-clés associés :

```
{
    "salutation": {
        "phrases": ["Bonjour", "Salut", "Coucou"],
        "mots_cles": ["bonjour", "salut", "hello"]
    },
    "recherche_internet": {
        "phrases": ["je veux faire une recherche internet", "peux-tu chercher
sur le net ?", "trouve ça en ligne"],
        "mots_cles": ["recherche", "internet", "chercher", "net"]
    },
    "recherche_video": {
        "phrases": ["je veux chercher une vidéo", "trouve une vidéo sur Youtube",
"montre-moi une vidéo"],
        "mots_cles": ["vidéo", "youtube", "trouve", "montre"]
    }
}
```

On voit dans le fichier `base_de_donnees.json` que les intentions sont définies à partir de phrases et de mots clés qui pourront permettre de les déceler.

3. Fonction d'analyse de mots-clés

Voici une fonction qui analyse la phrase de l'utilisateur et essaie de deviner son intention en fonction des mots-clés définis dans la base :

```
import json

def charger_base_de_donnees(fichier="base_de_donnees.json"):
    with open(fichier, "r", encoding="utf-8") as f:
        return json.load(f)

def analyser_intention(phrase, base_de_donnees):
    phrase = phrase.lower().strip()  # Mise en minuscule et suppression des
espaces inutiles

    # 1. Vérifier les phrases exactes
    for intention, contenu in base_de_donnees.items():
        phrases_exactes = contenu.get("phrases", [])

        if phrase in [p.lower() for p in phrases_exactes]:  # Vérifie si la
phrase correspond
            return intention

    # 2. Vérifier les mots-clés si aucune phrase exacte ne correspond
    for intention, contenu in base_de_donnees.items():
        mots_cles = contenu.get("mots_cles", [])

        for mot in mots_cles:
            if mot in phrase:
                return intention

    # Si aucune correspondance trouvée
    return phrase
```

Cette partie remplace notre ancienne fonction `comprendre_intention(phrase)` qui était bien plus basique.
On effectue un premier filtre afin de reconnaitre une phrase exacte, car c'est plus fiable, puis ensuite, s'il n'y a pas de reconnaissance, on effectue une recherche par mot clés.
S'il y a une reconnaissance, l'intention est retournée.
Dans le cas contraire, on retourne phrase et on passe à la phase d'ajout d'intention.

Nous allons aussi modifier l'ajout de nouvelles intentions avec le nouveau format utilisé dans le fichier `base_de_donnees.json`.
La fonc

Je vous fourni les fichiers modifiés.
Il est important de bien comprendre le code.

Ta mission va être d'enrichir ce que peut faire de code…