

Guide pour le Développement d'un Chatbot Médical en Santé

Ce guide présente les étapes clés pour développer un chatbot médical de base qui peut répondre à des questions simples en santé, comme les symptômes et les conseils de premiers soins.

Objectifs du Projet

1. **Développer un chatbot** capable de répondre à des questions basiques de santé.
2. **Se familiariser avec le traitement de langage naturel (NLP)** et les bases de l'apprentissage automatique (ML) dans un contexte médical.
3. **Explorer et analyser des données médicales (EDA)** pour construire une base de connaissances.

1. Technologies Recommandées

- **Python** : Langage de programmation principal.
- **NLP** :
 - **spaCy** : Bibliothèque de NLP pour le prétraitement du texte.
 - **BERT** (via **Transformers** de Hugging Face) : Pour une meilleure compréhension des questions.
- **Plateforme de Chatbot** :
 - **IBM Watson Assistant** ou **Dialogflow** pour gérer les intentions du chatbot.
- **Framework Web** (si souhaité) :
 - **Flask** pour intégrer une interface utilisateur basique.
- **Base de données** :
 - **SQLite** ou **MongoDB** pour stocker les historiques de conversation.

2. Étapes de Réalisation du Projet

Étape 1 : Collecte des Données et Exploration (EDA)

1. **Dataset Médical** : Trouvez un dataset adapté pour des questions simples de santé.
Exemples :
 - [MedQuAD](#) : FAQ médicales.
 - [CORD-19](#) : Informations sur la COVID-19.
2. **Analyse des Données (EDA)** :
 - Importez le dataset dans Python avec Pandas.
 - **Vérifiez les valeurs manquantes**, les doublons, et explorez les colonnes.
 - **Analyse de fréquence** : Comptez les types de questions les plus fréquents pour repérer les sujets populaires (ex: symptômes, conseils).

```
import pandas as pd

# Chargement des données
df = pd.read_csv('data/med_data.csv')
```

```
# Afficher des statistiques de base
print(df.info())
print(df['QuestionType'].value_counts())
```

Réflexion

- Quelles sont les questions les plus fréquentes ?
- Y a-t-il des catégories ou des thèmes qui apparaissent souvent dans les données ? (symptômes, premiers soins)

Étape 2 : Préparation des Données

1. Nettoyage des Données :

- Supprimez les caractères spéciaux et la ponctuation.
- Convertissez les textes en minuscules pour uniformiser le traitement.

```
import re

# Exemple de nettoyage
df['Question'] = df['Question'].str.lower().str.replace(r'[^a-zA-Z0-9]', '', regex=True)
```

2. Tokenisation et Lemmatization :

- Utilisez **spaCy** pour diviser les phrases en mots et pour la lemmatisation (simplification des mots).

```
import spacy
nlp = spacy.load("en_core_web_sm")

df['Question_tokens'] = df['Question'].apply(lambda x: [token.lemma_
for token in nlp(x) if not token.is_stop])
```

Étape 3 : Création des Intentions et Réponses

1. Définir les Intentions :

- Exemple d'intentions : Symptôme, Conseil, Traitement, etc.

2. Réponses Types :

- Pour chaque intention, rédigez des réponses simples et directes.
- Exemples :
 - Intention Symptôme : "Les symptômes de la grippe incluent la fièvre, la toux et les maux de tête."

3. Ajout des Synonymes et des Formulations :

- Enrichissez chaque intention avec différentes formulations de questions pour améliorer la compréhension du chatbot.

Étape 4 : Entraînement du Modèle NLP

1. Modèle BERT :

- Utilisez un modèle NLP pré-entraîné comme BERT pour une compréhension avancée.

2. Fine-Tuning :

- Entraînez le modèle pour qu'il s'adapte aux termes médicaux et questions spécifiques du dataset.

```
from transformers import pipeline
model = pipeline("question-answering")
```

Étape 5 : Développement et Intégration du Chatbot

1. **IBM Watson Assistant ou Dialogflow** :
 - Intégrez le modèle dans IBM Watson Assistant ou Dialogflow pour gérer les intentions et les réponses.
2. **Développer une Interface Simple** (optionnel) :
 - Utilisez Flask pour créer une interface web de test.

```
from flask import Flask, request, jsonify

app = Flask(__name__)

@app.route("/chatbot", methods=["POST"])
def chatbot():
    question = request.json.get("question")
    response = model({"question": question, "context": "Medical context..."})
    return jsonify(response)

app.run()
```

Étape 6 : Tests et Optimisations

1. **Tests Utilisateurs** :
 - Testez le chatbot avec différents types de questions.
2. **Ajustements** :
 - Modifiez les intentions, enrichissez les réponses et améliorez le modèle en fonction des retours.

3. Livrables

Code Source

- Code Python complet, incluant le nettoyage des données, le modèle NLP, et l'interface.

Documentation Technique

- **Architecture** du projet, instructions d'installation, et explications sur le fonctionnement du chatbot.

Manuel Utilisateur

- Guide d'utilisation pour interagir avec le chatbot.