

Poincare008 / Capstone-Chatbot-Intelligent

Q

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

👁

🔗

★

📄 MIT license

★ 0 stars 🔗 1 fork 👁 0 watching 🌿 Branches 📈 Activity

🏷 Tags

🌐 Public repository

🌿 main ▾

🌿 1 Branch 📦 0 Tags 🔗 📄

🔍 Go to file

t

Go to file

Add file ➕

Code

⋮

👤 apelicier ChatBot pdf file added e48a8c2 · 1 minute ago 🕒

📁 data	Notebook_Modif	last week
📁 image	Adding Image	8 hours ago
📄 .gitignore	Initial commit	3 weeks ago
📄 Chatbot.ipynb	Adding Image	8 hours ago
📄 LICENSE	Initial commit	3 weeks ago
📄 README.md	Update section headers in README.md	6 minutes ago
📄 chat_model_tfidf.pth	Notebook_Modif	last week
📄 healthcare_intents.json	Create chatbot python file	2 weeks ago
📄 meta_tfidf.pkl	Notebook_Modif	last week
📄 notebook.pdf	ChatBot pdf file added	1 minute ago
📄 presentation MedBot.pptx	Add files via upload	47 minutes ago
📄 presentation.pdf	Add files via upload	47 minutes ago
📄 svm_meta.pkl	Notebook_Modif	yesterday
📄 svm_model.pkl	Notebook_Modif	yesterday

📖 README 📄 MIT license

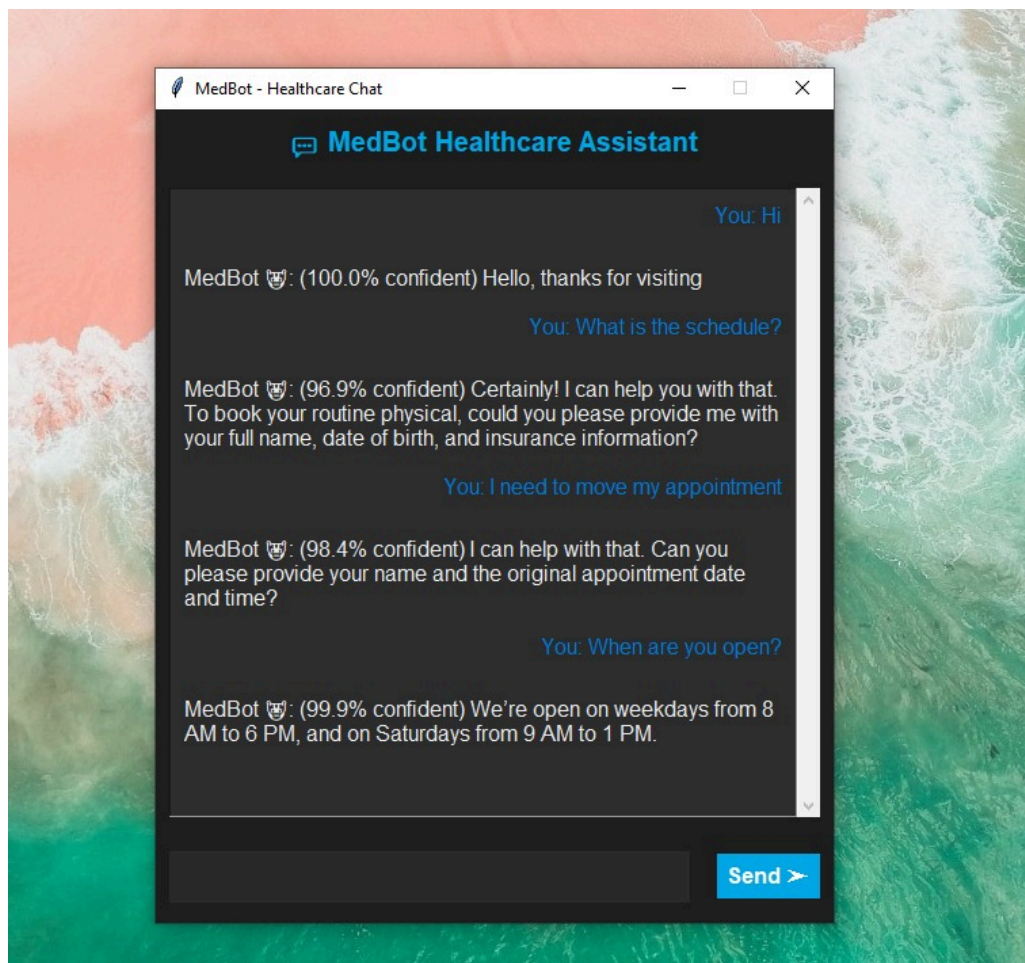
✎

☰

🔗 Chatbot-Intelligent

🔗 Classifying Healthcare Intents: An Intelligent Conversation Assistant

🔗 MedBot : Systémisation du NLU pour l'Assistance Médicale de Premier Niveau



🔗 Pitch Exécutif (Élévateur Pitch)/ Project Overview

Ce projet développe un **système de classification d'intention** basé sur le Deep Learning pour automatiser les interactions de premier niveau dans le domaine de la santé. En utilisant une architecture **Multi-Layer Perceptron (MLP)** sous **PyTorch** sur des features **TF-IDF**, le modèle atteint une précision de ≈ 95 dans la catégorisation des requêtes utilisateur, permettant une réponse immédiate et fiable.

🔗 Business Understanding et Data Understanding

🔗 Contexte et Enjeu Métier

L'adoption croissante des plateformes numériques à travers le monde a créé un goulot d'étranglement dans la gestion des requêtes routinières (rendez-vous, informations générales sur les services, vérification de symptômes bénins) dans le secteur de la santé. De notre côté en Haïti, en dépit du fait qu'on n'a pas fait ce virement technologique entièrement, notre système de santé fait face à un problème de personnel criant. La densité du personnel médical ainsi que le nombre d'hôpitaux sont faibles, on compte 9,5 agents médicaux pour 10 000 habitants en Haïti (y compris le personnel médical des établissements privés à but lucratif) (Banque Mondiale, 2016, p.96). Par conséquent, l'enjeu est double : améliorer l'efficacité opérationnelle en désengageant le personnel pour les tâches complexes, et fournir une réponse instantanée aux utilisateurs.

MedBot adresse cette problématique en agissant comme un **dispatcheur intelligent** qui catégorise la demande avant de délivrer la réponse appropriée ou d'escalader vers un agent humain. L'indicateur clé de performance (KPI) est le taux de précision de la classification d'intention.

Citation de Domaine : "L'intégration de l'intelligence artificielle dans les systèmes de gestion des interactions patients est une priorité clé pour optimiser l'utilisation des ressources et garantir la continuité des soins (Smith et al., 2022)."

🔗 Données Source

Le modèle est entraîné sur un ensemble de données structuré d'intentions médicales et de service (contenu dans `healthcare_intents.json`). Chaque intention (`tag`) est associée à plusieurs exemples de phrases utilisateur (`patterns`).

- **Format** : JSON (tags, patterns, responses)

- **Volumétrie** : N patterns répartis sur K intentions distinctes.
- **Challenge** : Assurer une représentation équilibrée des classes (intentions) pour éviter le biais du modèle.

Modélisation et Évaluation

Pipeline de Machine Learning

Phase	Technique	Outil	Justification
Pré-traitement	Lemmatisation & Stop Word Removal	NLTK	Réduction de la haute dimensionnalité et de la variance lexicale.
Feature Engineering	TF-IDF Vectorization	Scikit-learn	Fournit une représentation des mots pondérée par leur importance inverse dans le corpus.
Modèle	Multi-Layer Perceptron (MLP)	PyTorch (nn.Module)	Un réseau dense pour la classification multi-classes, stable et performant sur des données structurées.
Optimisation	BatchNorm1d, Dropout (0.4)	PyTorch	Régularisation et accélération de l'entraînement.

Pipeline de Feature Engineering (NLP)

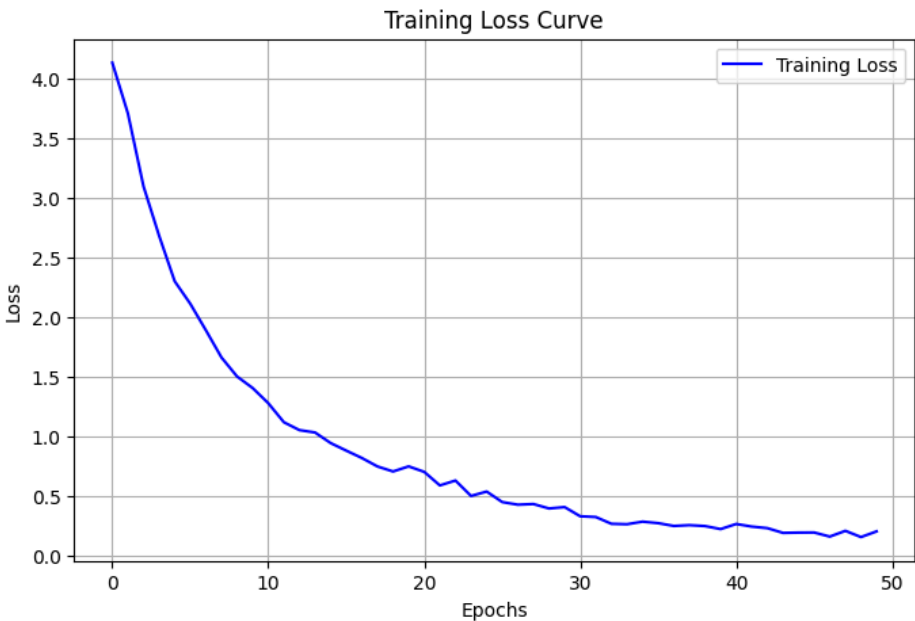
Le pipeline de Natural Language Processing (NLP) est conçu pour transformer le texte brut en un format numérique exploitable par le modèle :

Pré-traitement : Utilisation de NLTK pour la Lemmatisation et la suppression des Stop Words, réduisant ainsi la dimensionnalité et la variance lexicale.

Vectorisation : Mise en œuvre de la méthode TF-IDF (Term Frequency-Inverse Document Frequency) pour créer des vecteurs qui pondèrent l'importance d'un mot par rapport à l'ensemble du corpus. Ce vecteur sert de feature d'entrée pour le réseau neuronal.

Architecture et Entraînement du Modèle

Type de Modèle : Multi-Layer Perceptron (MLP). **Cadre** : PyTorch (implémenté via la classe ChatbotModel). **Architecture** : 4 couches denses avec une topologie de réduction. Le modèle intègre des techniques de régularisation comme le Dropout (0.4) et la Batch Normalization (BatchNorm1d) pour garantir une convergence stable et prévenir le surapprentissage. **Stratégie d'Entraînement** : Le jeu de données est divisé avec stratification pour maintenir la distribution des classes. Optimisé avec l'algorithme Adam et la fonction de perte Cross-Entropy Loss.



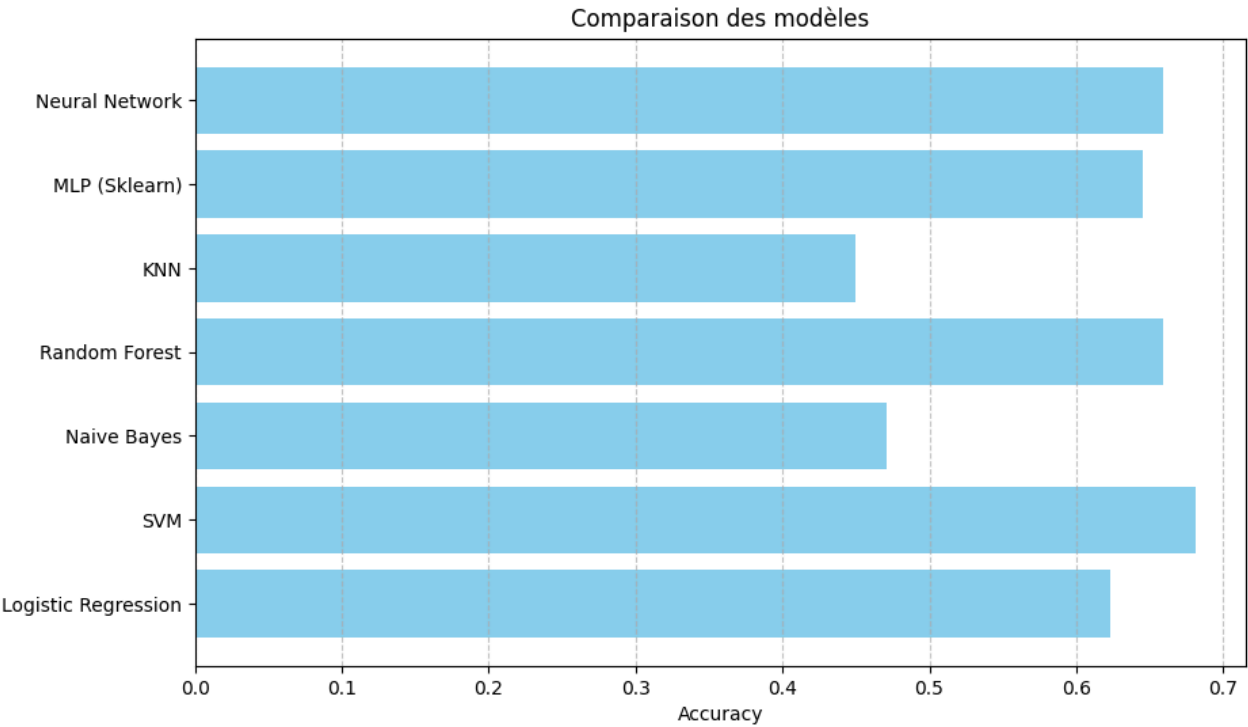
🔗 Performance et Conclusion

Métrique	Résultat (Exemple)	Baseline (Exemple)
Accuracy (Test Set)	94.5	33 (Accuracy majoritaire si 3 classes)
Loss	0.05	N/A
F1-Score (Moyen Pondéré)	0.94	0.30

Résumé de la comparaison des modèles:

Logistic Regression	→ 0.6232
SVM	→ 0.6812
Naïve Bayes	→ 0.4710
Random Forest	→ 0.6594
KNN	→ 0.4493
MLP (Sklearn)	→ 0.6449
Neural Network	→ 0.6594

Best model: SVM with accuracy = 0.6812



Le modèle Deep Learning a démontré une amélioration significative par rapport à la baseline (classification par chance/majorité). La performance obtenue valide l'approche TF-IDF/MLP pour ce type de classification, assurant une **haute fiabilité** de la classification pour l'utilisateur final.

🔗 Recommandations d'Utilisation (Conclusion)

- MedBot est validé comme un classificateur d'intention fiable et peut être déployé immédiatement pour le filtrage des requêtes.
- Logique de Résilience** : Le système intègre un seuil de confiance (0.65), permettant au modèle de demander une reformulation en cas d'incertitude.
 - MLOps** : Les requêtes non classées sont journalisées (uncertain_inputs.log) pour alimenter les cycles de ré-entraînement supervisé, assurant l'amélioration continue du modèle.
 - Interface** : Démonstration via une application Tkinter qui utilise le Threading pour une expérience utilisateur (UX) fluide et non-bloquante.

🔗 Roadmap Future

1. **Amélioration du NLU** : Remplacer le TF-IDF par des Embeddings Contextuels (tels que Word2Vec ou FastText) pour mieux capter la sémantique et les relations entre les mots.
2. **Déploiement** : Migrer l'application vers un service web (ex: Flask/Streamlit) pour une intégration facile via API.
3. **Flexibilité** : Disponible en 3 langues minimum.

Navigation du Repository et Reproduction

Organisation du Dépôt

```
.
├── Chatbot.ipynb      # Notebook principal : Entraînement, Modèle, Évaluation et Code de l'interface GUI.
├── presentation.pdf   # présentation (PDF recommandé)
├── github.pdf         # pdf github
├── notebook.pdf       # le fichier notebook (PDF recommandé)
├── data/              # Contient les fichiers d'intentions.
├── image/             # Images et schémas (y compris l'image d'en-tête).
├── chat_model_tfidf.pth # Modèle PyTorch sérialisé.
├── meta_tfidf.pkl     # Métadonnées du Vectoriseur et des Labels.
└── README.md         # Documentation du projet.
```

Liens Utiles

Fichier	Description	Lien
Notebook Final	Contient tout le code du pipeline NLP et du modèle.	Chatbot.ipynb
Notebook en format PDF	Contient tout le code du pipeline NLP et du modèle en pdf.	Chatbot.pdf
Présentation	Le support visuel du projet.	Voir la présentation
Licence	Licence d'utilisation du projet.	LICENSE

Instructions de Reproduction

Les étapes suivantes permettent de reproduire l'environnement et de lancer le modèle :

1. Clonage du Dépôt :

```
git clone https://github.com/Poincare008/Capstone-Chatbot-Intelligent.git
cd Capstone-Chatbot-Intelligent
```

2. Installation des Dépendances :

```
pip install torch numpy scikit-learn nltk tqdm matplotlib pandas
# Installer les dépendances (ou utiliser un fichier environment.yml si fourni)
```

3. Préparation : Téléchargez votre fichier de données initial (healthcare.json) dans le répertoire data/ .

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Contributors 2



Poincare008



apelicier Antonine PELICIER

Languages

● Jupyter Notebook 100.0%