Rapport du Projet Chatbot ISET

Projet de Machine Learning - DSIR 1

Année Universitaire 2024/2025

# Table des Matières

* 1. Introduction
* 2. État de l'Art
* 3. Architecture du Système
* 4. Implémentation Technique
* 5. Fonctionnalités Innovantes
* 6. Évaluation et Résultats
* 7. Conclusion et Perspectives
* 8. Annexes

# 1. Introduction

## 1.1 Contexte

Le projet de chatbot ISET s'inscrit dans le cadre du cours de Machine Learning de la formation DSIR 1. L'objectif est de développer un assistant virtuel capable de guider les utilisateurs du site web de l'ISET en répondant à leurs questions et en les orientant vers les pages pertinentes.

## 1.2 Objectifs

* Développer un chatbot intelligent pour le site de l'ISET
* Répondre aux questions des étudiants sur divers sujets
* Optimiser la navigation des utilisateurs
* Implémenter des techniques de NLP et de Machine Learning

## 1.3 Problématique

La principale problématique est de créer un système capable de :

* Comprendre les questions en langage naturel
* Fournir des réponses pertinentes
* Guider les utilisateurs vers les bonnes pages
* S'adapter aux différentes formulations des questions

# 2. État de l'Art

## 2.1 Approches Existentes

**Chatbots basés sur des règles**

* Avantages : Simple, prévisible
* Inconvénients : Rigide, maintenance coûteuse

**Chatbots basés sur le Machine Learning**

* Avantages : Flexible, adaptable
* Inconvénients : Nécessite des données d'entraînement

**Chatbots hybrides**

* Avantages : Combine les forces des deux approches
* Inconvénients : Plus complexe à maintenir

## 2.2 Techniques de NLP

**Prétraitement du texte**

* Tokenization
* Suppression des stopwords
* Stemming/Lemmatisation

**Représentation vectorielle**

* TF-IDF
* Word Embeddings
* Bag of Words

**Modèles de classification**

* K-Nearest Neighbors
* Support Vector Machines
* Naive Bayes

# 3. Architecture du Système

## 3.1 Composants Principaux

**Module de Prétraitement**

* Nettoyage du texte
* Tokenization
* Gestion des stopwords
* Stemming

**Module de Machine Learning**

* Vectorisation TF-IDF
* Classification KNN/SVM
* Calcul de confiance

**API Flask**

* Gestion des requêtes
* Prétraitement
* Prédiction
* Réponses

**Interface Streamlit**

* Interface utilisateur
* Gestion des conversations
* Affichage des résultats

# 4. Implémentation Technique

## 4.1 Exemple de Code

class TextProcessor:  
 def \_\_init\_\_(self, language: str = 'french'):  
 self.language = language  
 self.stemmer = SnowballStemmer(language)  
 self.stop\_words = set(stopwords.words(language))  
  
 def process(self, text: str) -> List[str]:  
 cleaned\_text = self.clean\_text(text)  
 tokens = self.tokenize(cleaned\_text)  
 tokens = self.remove\_stopwords(tokens)  
 tokens = self.stem(tokens)  
 return tokens

# 5. Fonctionnalités Innovantes

**Système de Confiance**

* Calcul du score de confiance pour chaque réponse
* Affichage visuel du niveau de confiance
* Gestion des cas de faible confiance

**Suggestions Proactives**

* Analyse du contexte de la conversation
* Proposition de questions connexes
* Liens vers des pages pertinentes

**Gestion Multilingue**

* Support du français et de l'anglais
* Détection automatique de la langue
* Prétraitement adapté à chaque langue

# 6. Évaluation et Résultats

## 6.1 Métriques d'Évaluation

* Précision des réponses
* Temps de réponse
* Satisfaction utilisateur
* Taux de résolution

# 7. Conclusion et Perspectives

## 7.1 Bilan du Projet

* Objectifs atteints
* Challenges rencontrés
* Solutions apportées

## 7.2 Limites Actuelles

* Taille du dataset
* Complexité des questions
* Gestion des cas particuliers

## 7.3 Améliorations Futures

* Enrichissement du dataset
* Ajout de nouvelles fonctionnalités
* Optimisation des performances

# 8. Annexes

## 8.1 Structure du Projet

iset-chatbot/  
├── data/ # Données d'entraînement  
├── models/ # Modèles entraînés  
├── src/  
│ ├── preprocessing/ # Prétraitement  
│ ├── models/ # Modèles ML  
│ ├── api/ # API Flask  
│ └── utils/ # Utilitaires  
├── tests/ # Tests  
├── web/ # Interface  
└── config/ # Configuration