



Sciences des Données Appliquées

Rapport de projet

ChatBot industriel

<https://github.com/Abd-MOUSSAOUI/Chatbot-printer>

par

Abderahmane Moussaoui (amoussaoui@etu.unistra.fr)

Master 2 SDSC (2020 - 2021)

Introduction

Une entreprise d'impression souhaite développer son activité d'impression en ligne. Afin d'éviter le recrutement d'un commercial et d'un développeur web, l'entreprise compte acquérir une imprimante munie de plusieurs capteurs ou faire construire une application ChatBot qui récoltera les demandes et ordonnance l'impression des documents par une approche First In First Out (FIFO).

L'objectif de ce projet est de développer un code de prise en compte automatique de demande d'impression par « ChatBot » et prévoir l'ordonnancement des tâches d'impression.

Approche utilisée

Partie hors-ligne:

Pour la 1er partie du projet, j'ai commencé à créer les données et les préparer pour l'entraînement. J'ai choisis d'ajouter "salutation" et "aide" à "impression" et "autres" donc mes données d'entrainements contiennent 4 catégories..

J'ai utiliser la méthode de **Bag Of Words**, à l'aide de la library nltk, donc pour cela j'ai collecter tous les données ou plutôt les mots de mes train data pour pouvoir utiliser la méthode BOW, je coupe (split) le message en liste de mots ou tokens (tokenize) et si le mot existe il prends la valeur 1 sinon 0. J'ai aussi utilisé un stemming qui consiste à générer la racine des mots (enlever la fin du mots) qui peut servir dans le cas d'un verbe conjugué ou pas par exemple..

La 2eme partie, j'ai créé le model qui a été réalisé à l'aide de la library Pytorch, c'est un model simple avec 3 couches Linear, une fonction d'activation Relu qui s'active après la 1er et la 2eme couche, CrossEntropyLoss pour loss et Adam pour l'optimisation.

le modèle a été entraîné avec les données que j'ai préparé dans la partie 1, (bag of words en entrée) 2 couche caché et une catégorie de message en sortie...

A la fin de ce traitement je stock le statut du model et des données importante (input_size, output_size,...) dans un fichier "data.pth" (Pytorch file) pour pouvoir l'utiliser par le chatbot par la suite.

Partie en-ligne:

Dans la 3eme partie, j'ai créé le ChatBot qui récupère les données et le modèle stocker dans le fichier data.pth premièrement et après entre dans la boucle while qui communique avec les utilisateurs, le ChatBot récupère le message de l'utilisateur le prépare et le transforme en BOW (tokenize, bag_of_words..) et finalement il le passe au model pour prédire à quel catégorie appartient le message ("salutation", "impression", "aide" ou "autre"), et il réponds randomly avec une des réponses pré-définis pour chaque catégorie si sa prédiction est supérieur ou égale à 0.7 (70%) sinon il informe l'utilisateur qu'il n'a pas compris et il lui propose de demander de l'aide.

Tous les messages reçu seront stocker dans un fichier "report.txt" avec la date et l'heure du message et la catégorie que le model à prédit.

Si le message reçu est de la catégorie impression, je vérifie que le message contient le nom et le nombre de pages à imprimer pour le prendre en considération, sinon je demande à l'utilisateur de refaire sa

demande en indiquant le nom du fichier qui doit commencer par “doc..” et le nombre de pages qu’il contient.

Dans le cas où le message contient le nom et le nombre de page du documents, je le prends en considération et j’ajoute la demande d’impression dans le fichier “agenda.txt” avec la date l’heure du début et de fin d’impression et le nom du documents.

L'ordonnancement des demandes est géré aussi par le ChatBot avec la règle FIFO, donc plusieurs utilisateur peuvent utiliser le ChatBot en même temps. J’ai utilisé un fichier “counter.txt” qui stock l’heure de la fin de la dernière impression.. .

On considère que le service d’impression travaille 24/24, je n’ai pas rajouté des contraintes du non disponibilité de l’imprimante.. donc le ChatBot prends les commandes 24/24 et organise l’impression du document après la fin de l’impression du dernier doc dans la liste.

Mode d’emploi

Requirement

```
pip3 install nltk numpy torch
```

Usage

Il faut entrainer le model avant d'utiliser le ChatBot, pour cela taper:

```
python3 train.py
```

Maintenant vous pouvez lancer le ChatBot:

```
python3 chatbot.py
```

Lien GitHub: (le projet est en privé, il passera en public à minuit)

<https://github.com/Abd-MOUSSAOUI/Chatbot-printer>