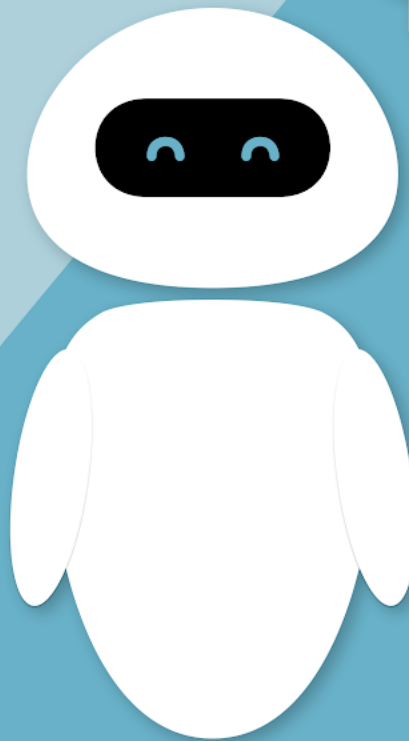


CHATBOT

Alvarez Luka, Calunsag Sabina



Hi ! What's your
favorite movie ?



September 2020, V 1.0



TABLE DES MATIÈRES

INTRODUCTION	3
Description	3
Pourquoi avoir choisi ce sujet ?	3
Quel logiciel de développement ?	3
Pourquoi python ?	3
CAHIER DES CHARGES	4
But	4
Fonctionnalités à réaliser	4
Outils	4
Travail à rendre	4
ANALYSE FONCTIONNELLE	5
Interface	5
RAPPORT DE TESTS	5
CONCLUSION	5
ANNEXES	6
Plannings	6
Références	7
Autre	7

INTRODUCTION

Description

Nous allons faire une IA qui analyse les messages envoyés et répond aux questions des utilisateurs en rapport avec le cinéma qui utilise un dataset contenant des données sur des films.

Dans un prochain temps, réussir à changer / ajouter la dataset du bot pour qu'il apprenne nos goûts sur le cinéma.

Pourquoi avoir choisi ce sujet ?

Étant des informaticiens, nous sommes au courant de l'importance de l'IA dans le monde actuel. Aujourd'hui il y a des IA qui sont utilisées partout. C'est pour cela que nous avons choisi ce projet pour mieux comprendre le fonctionnement des IA de manière générale.

Quel logiciel de développement ?

Pour programmer ce chatbot en python nous avons décidé d'utiliser visual studio code. Il permet de facilement intégrer git directement dans l'éditeur. En plus vscode est facilement modulable grâce à ses extensions.

Pourquoi python ?

Python est un langage de programmation facile à apprendre et très performant...

CAHIER DES CHARGES

But

Le but de ce projet est d'élargir notre connaissance sur le fonctionnement d'un chatbot et de réussir à lui apprendre des choses à partir d'un modèle de données. À la fin du projet, on va pouvoir interagir avec le chatbot en utilisant une interface sur web.

Fonctionnalités à réaliser

- Setup l'environnement
- Tester le projet
- Reproduire le projet
- Changer de modèle de donnée
- Entraîner le chatbot
- Faire une interface graphique

Outils

- PyTorch
- Python
- Librairies
- Balsamiq

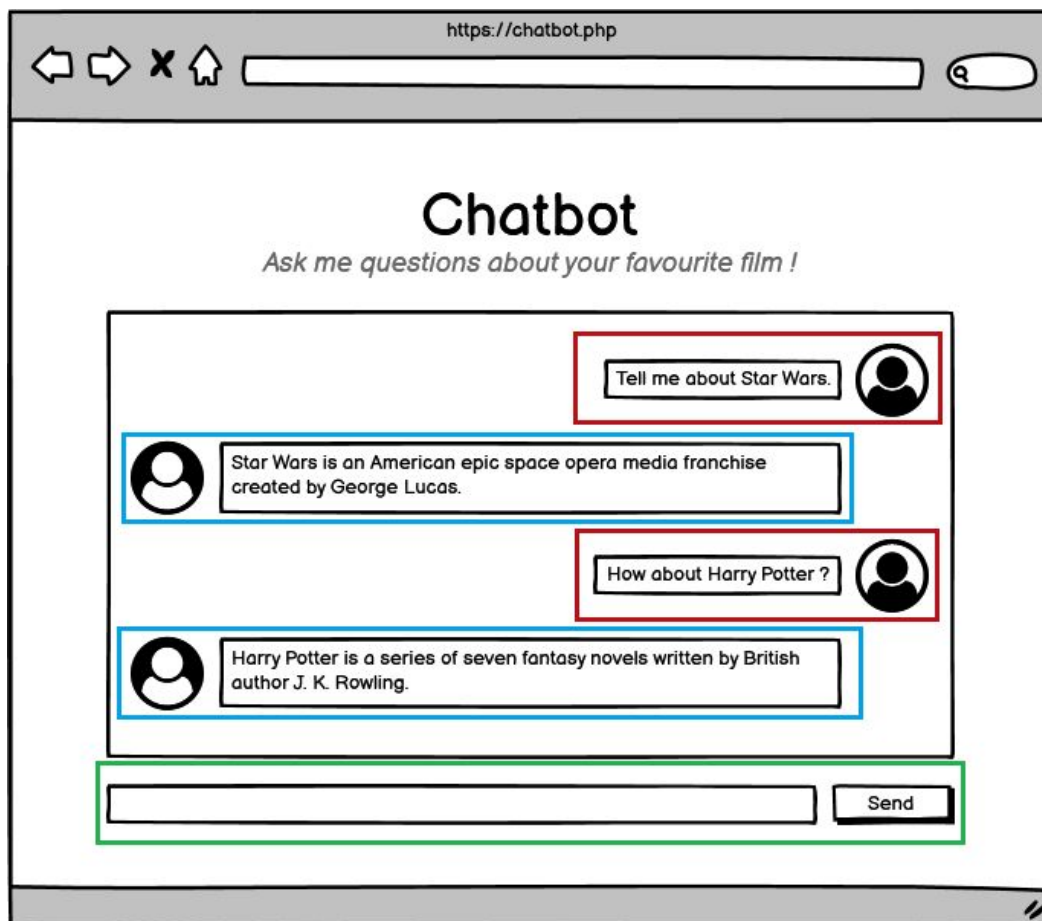
Travail à rendre

ANALYSE FONCTIONNELLE

Interface

On peut voir sur l'image ci-dessous, la page principale qui permet à l'utilisateur de communiquer avec le chatbot à l'aide d'un message box.

- **Rouge** : Les messages envoyés par l'utilisateur avec un icône à droite.
- **Bleu** : Les messages envoyés par le chatbot(IA) avec un icône à gauche.
- **Vert** : Un textbox pour insérer un message puis un bouton à droite pour l'envoyer.



Fonctionnalités principales

- Pouvoir communiquer avec le chatbot en utilisant le textbox.
- Lui apprendre par la suite d'autres sujets en utilisant une librairie différente.

RAPPORT DE TESTS

Début du projet

Nous avons utilisé le langage python et le framework Pytorch pour la réalisation de notre projet chatbot. Nous, qui connaissons très peu sur le fonctionnement et le développement d'un chatbot, nous avons décidé de partir sur un tutoriel puis par la suite intégrer d'autres librairies dans le projet pour lui apprendre.

Pour mettre en place cela, nous avons d'abord installé Pytorch et en suivant le tutoriel nous avons rencontrés quelques erreur :

Error :

```
FileNotFoundError: [Errno 2] No such file or directory:
'./save/training_data/corpus/voc.tar' During handling of the above exception,
another exception occurred:
=> create directory : save/training_data/corpus/voc.tar
```

```
RuntimeError: Attempting to deserialize object on a CUDA device but
torch.cuda.is_available() is False. If you are running on a CPU-only machine,
please use torch.load with map_location=torch.device('cpu') to map your
storages to the CPU.
```

```
line 488 => map_location=None => map_location='cpu'
file => /home/luka/.local/lib/python3.7/site-packages/torch/serialization.py
```

Après avoir réglé les problèmes, nous sommes passés par le training, c'est-à-dire envoyer des données au chatbot pour qu'il apprenne au fur à mesure.

Training:

































```
python3 main.py -tr ./data/corpus2.txt./data/corpus2.txt
-l./save/model/movie_subtitles/1-1_512/50000_backup_bidir_model.tar -lr
0.0001 -it 50000 -b 64 -p 500 -s 1000
```

































Intégration d'une autre librairie



Par la suite, nous allons essayer d'intégrer une librairie différente de celle du tutoriel.

ANNEXES

Plannings

	Planning Prévisionnel 1								
	31.09.2020	07.09.2020	14.09.2020	21.09.2020	28.09.2020	05.10.2020	12.10.2020	19.10.2020	26.10.2020
Conception	 	 							
Poster			 						
Cahier des charges				 					
Développement					 	 	 	 	 
Interface									
Documentation			 	 	 	 	 	 	 

	Planning Prévisionnel 2								
	02.11.2020	09.11.2020	16.11.2020	23.11.2020	30.11.2020	07.12.2020	14.12.2020	21.12.2020	
Conception									
Poster									
Cahier des charges									
Développement	 	 	 	 	 	 			
Interface							 	 	
Documentation	 	 	 	 	 	 	 	 	

Légende :	Luka		Sabina	
-----------	------	---	--------	---

Références

<https://code.visualstudio.com/docs/python/python-tutorial>

<https://github.com/ywk991112/pytorch-chatbot/tree/master>

Autre

Error :

FileNotFoundError: [Errno 2] No such file or directory:

'./save/training_data/corpus/voc.tar'

During handling of the above exception, another exception occurred:

=> create directory : save/training_data/corpus/voc.tar

RuntimeError: Attempting to deserialize object on a CUDA device but torch.cuda.is_available() is False. If you are running on a CPU-only machine, please use torch.load with map_location=torch.device('cpu') to map your storages to the CPU.

line 488 => map_location=None => map_location='cpu'

file => /home/luka/.local/lib/python3.7/site-packages/torch/serialization.py

Training:

python3 main.py -tr ./data/corpus2.txt./data/corpus2.txt -l

./save/model/movie_subtitles/1-1_512/50000_backup_bidir_model.tar -lr 0.0001 -it

50000 -b 64 -p 500 -s 1000