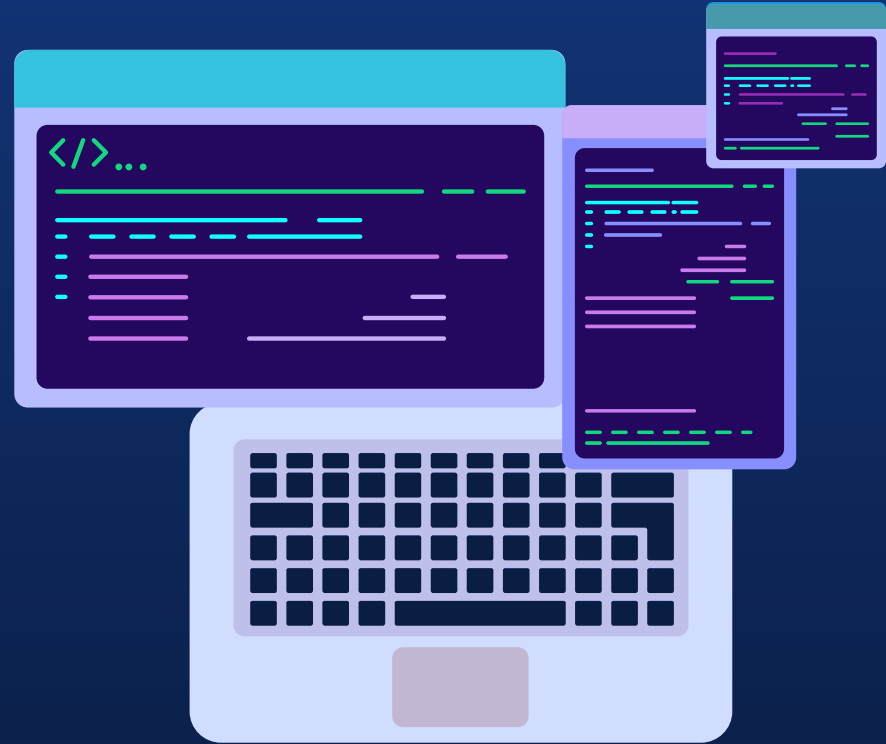


T-AI-901



PLAN

01

INTRODUCTION

02

ORGANISATION DE
L'ÉQUIPE

03

ARCHITECTURE DU
PROJET

04

RESULTATS &
PERFORMANCE

05

DEMO






01



INTRODUCTION

- Effectuer une recherche de trajet entre les gares ferroviaires de France
 - Par message vocal , fichier audio ou par écrit
 - Proposer l'itinéraire le plus optimal
- 

02

ORGANISATION DE L'ÉQUIPE

- Pseudo Agile
- Projet R&D
- Trello
- Projet découpé en plusieurs modules



ÉQUIPE

Bastien

Valentin

Axel

Fabrice

Jonathan

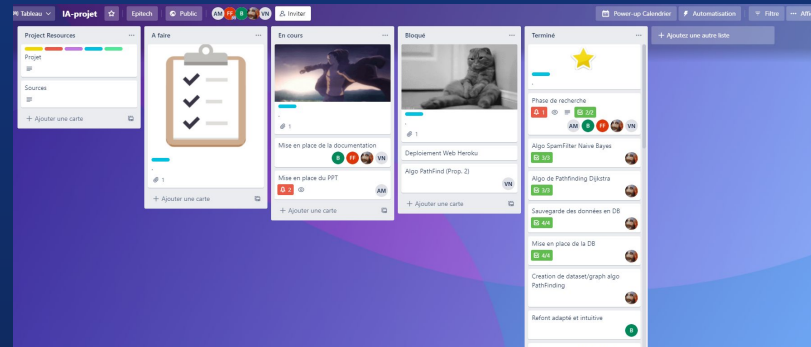
Organisation de l'équipe

1 - "Pseudo-Agile"

2 - Utilisation d'un trello et de Teams

3 - Découpage en plusieurs module :

- Front-end
- Back-End
- Speech-to-Text
- Spam-Filter
- NLP
- Path-Finding



03 ARCHITECTURE DU PROJET

- Speech-to-text
- Spam-filter
- Natural Language Processing
- Path-Finding
- Front-end & Back-end





Speech To Text - Google Cloud Speech API

- 1 - Mieux adapté à la langue Française
- 2 - Permet de filtrer le message malgré les bruits en fond sonores
- 3 - Implémentation et utilisation facile



**Google Cloud
Speech API**

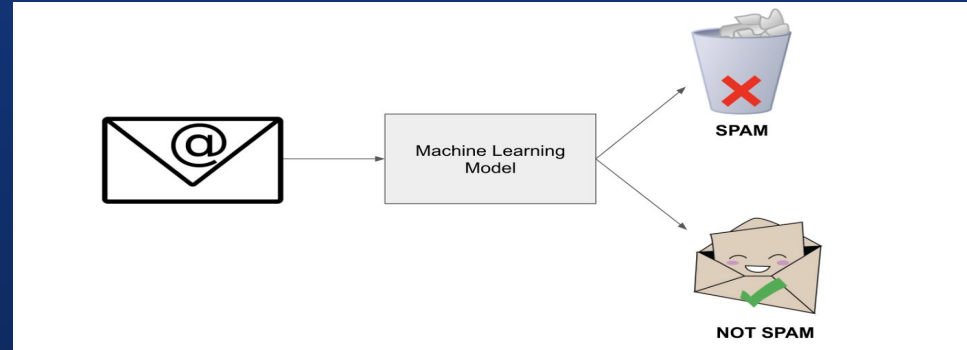


Spam-Filter

1 - Inspiré du filtrage d'emails

2 - Permet de décider si le texte présenté est valide ou pas

3 - Afficher un message d'erreur personnalisé si le texte n'est pas valide



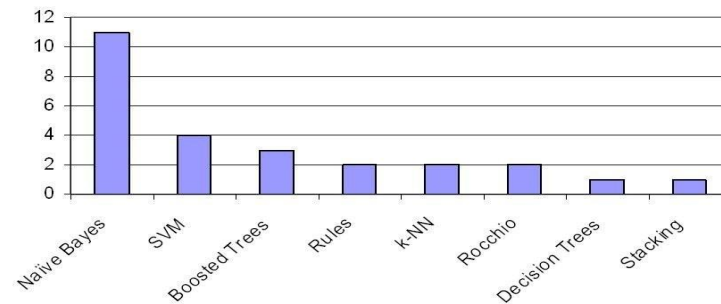
Spam-Filter - Solutions Possible

1 - Naive Bayes , basé sur la probabilité d'apparition d'un mot désirable/indésirable dans un texte

2 - Simple jeu de règles prédéfinies, avec liste noire. Algo limité car règles souvent simple à contourner

3 - Modèles KNN

Algorithms Used in Spam Filtering





Spam-Filter - Solutions retenue : Naïve Bayes Classifier

- 1 - Modèle populaire et efficace, règles prédéfinie robuste et modulable
- 2 - Création d'un dataset 50% spam et 50% ham
- 3 - Phase d'entraînement , enregistré en DB
- 4 - Application des règles du spam filter sur le texte
- 5 - Evaluation du modèle avec la métrique d'accuracy

Naïve Bayes Classifier

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$



Thomas Bayes
1702 - 1761



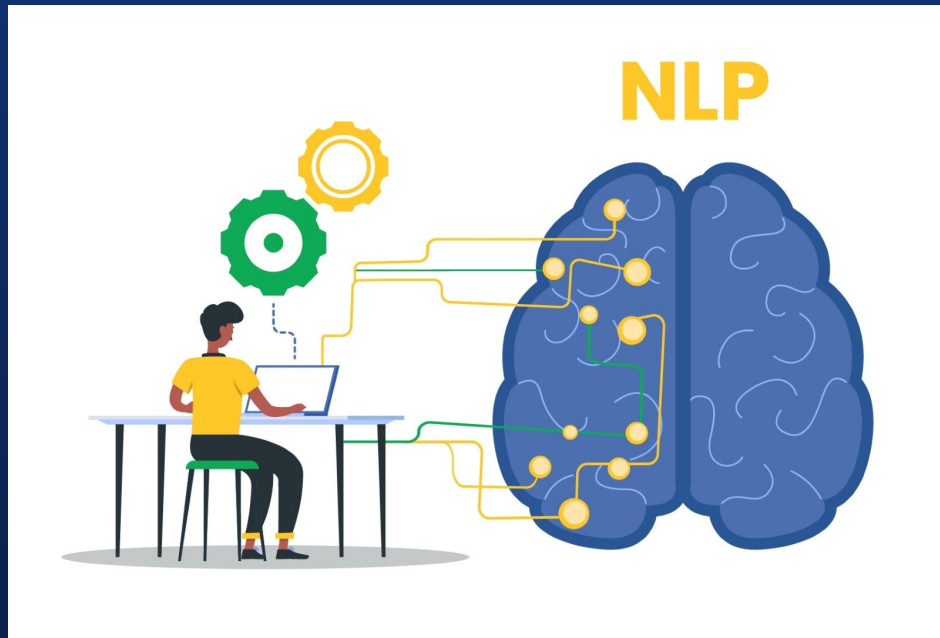
Natural Language Processing

Différentes approches

- NLP (Natural Language Processing)
- NLU (Natural Language Understanding)
- NLG (Natural Language Generation)

Solution retenue NLP avec Spacy

- Patterns - matchers

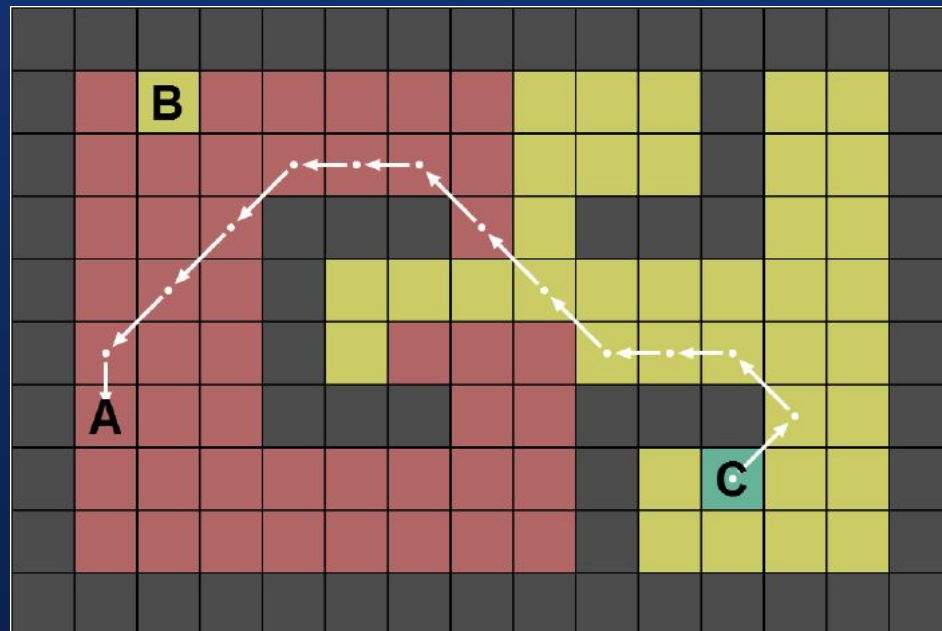




Path-Finding

1 - Proposer à l'utilisateur l'itinéraire ferroviaire le plus optimal

2 - Deux options possible : Dijkstra ou A*



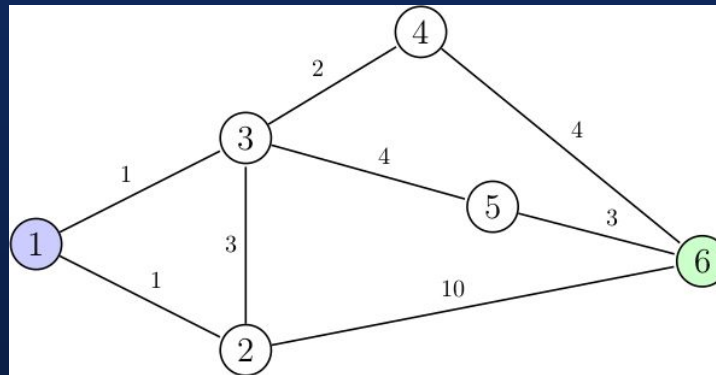
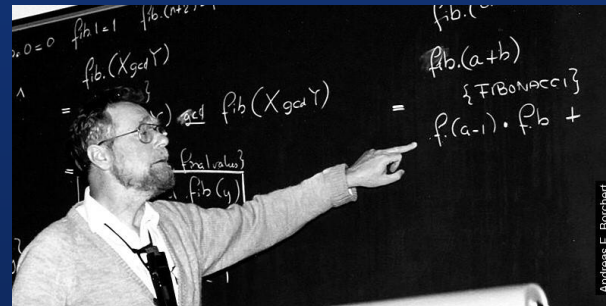


Path-Finding - Solution retenue : Dijkstra

1 - Par manque de temps , la solution A* à été mise de côté pour la solution Dijkstra

2 - Conscient que la solution A* était la plus approprié pour ce problème

3 - Le dataset



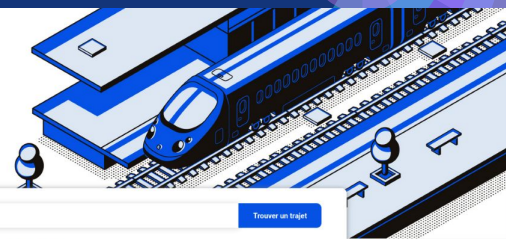
FRONT-END

1 - Mise en place d'une Single Page Application

2 - Peut soumettre sa requête à l'écrit , à l'oral ou via un fichier audio



LE CHEMIN LE PLUS
COURT VERS VOTRE
DESTINATION.

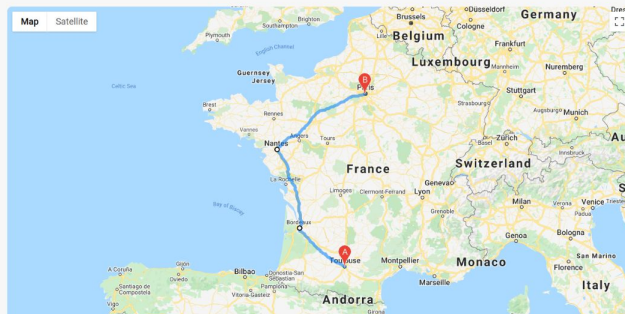


Où allez-vous ?

Trouver un trajet



- Gare de Toulouse Matabiau, Toulouse
- Gare de Bordeaux-Mérignac, Bordeaux
- Gare de Nantes, Nantes
- Gare de Paris-Montparnasse, Paris



BACK-END

Implémentation d'une API en Flask





04 RESULTATS & AMELIORATIONS



Resultats

1 - NLP

- Phrases de tests validées : 20/22
- Limites de Spacy sur la compréhension du contexte

2 - Spam filter

- 92% accuracy sur 3600 messages test






Aller plus loin

Coté Path-Finding

- Prise en compte du temps entre correspondance
- Afficher liste gare si la demande n'est pas explicite
- Avoir un deuxième modèle de comparaison (A^*)

Coté Spam-Filter

- Sauvegarder les messages clients en base de données
 - Détection IP de la ville de départ
 - Sens de la phrase perdue
- 



Aller plus loin

Application dans sa globalité

- Héberger la solution sur Heroku
- Rendre la solution polyglotte



HEROKU



DEMO TIME



MERCI

Avez vous des
questions ?

