UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

STNP solution IA

Documentation technique

Nom du projet	STNP Solution IA
Type de document	Documentation technique
Date	24/03/2022
Version	1.3

Table des matières

Résumé du document

Ce document technique est la présentation des différentes parties du projets avec ses fonctionnalités, elle est divisé en 2 parties :

- La partie frontend, l'interface
- La partie backend, la configuration et les fonctionnalités

Rappel sur le fonctionnement de l’application

Description de l’application

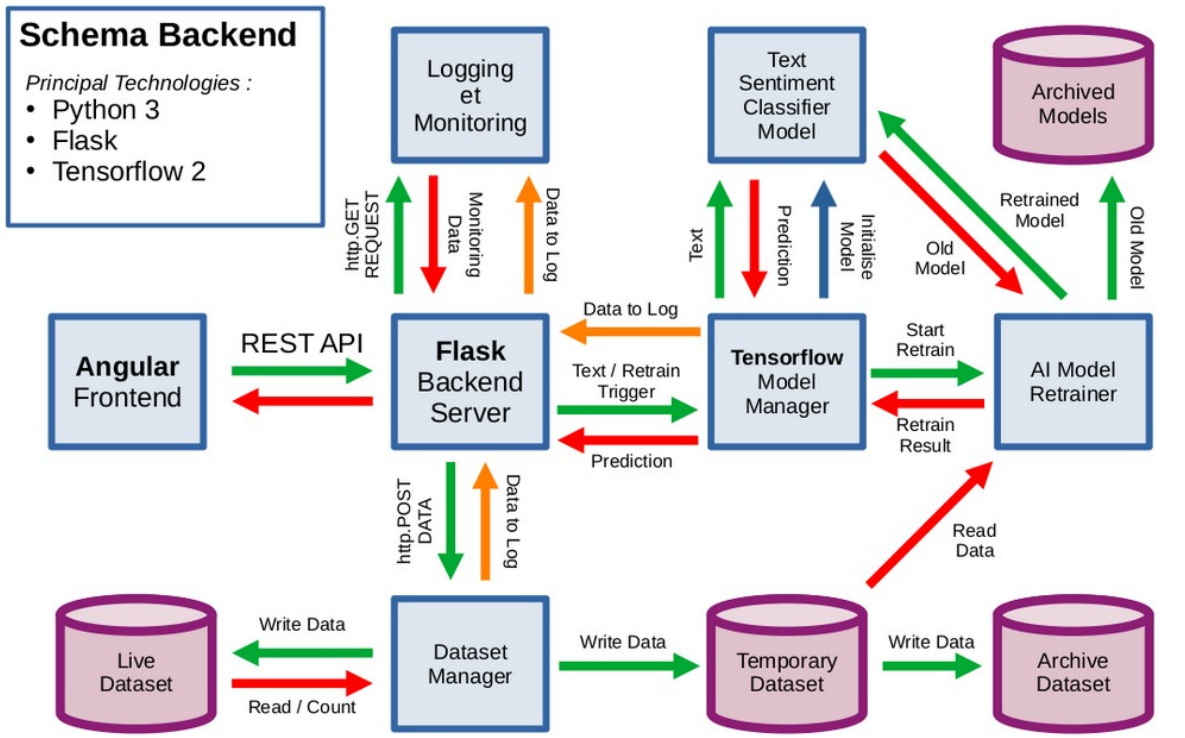
Notre projet consiste à créer une application web, qui prendra les commentaires des utilisateurs, puis déterminera si celui-ci est un commentaire positif ou négatif.
Nous aurons une partie utilisateur, où il y aura une zone de texte pour saisir une phrase en anglais. Et, une partie administrateur où l’on verra le nombre de commentaires positifs et négatifs qui auront été saisis.
Pour savoir le sentiment du commentaire, nous avons un jeu de données où nous avons effectué un apprentissage. Tous les 1 000 commentaires, un nouvel apprentissage est fait.

Décomposition du projet

Notre projet se décompose en 2 grandes parties :

- Le frontend, qui sera l’interface graphique pour les utilisateurs et l’administrateur.
- Le backend comportera toutes les fonctionnalités reliées à l’interface.

Architecture globale



Architecture de l’application

Le frontend

Technologies utilisées

L’interface graphique a été effectuée avec le framework Angular open-source avec TypeScript.

Le backend

Technologies utilisées

Le backend est codé en python, un langage de programmation interprété. Puis, nous avons utilisé le micro framework flask, qui est un framework open-source de développement web en python, d'où le langage choisi. De plus, nous avons utilisé TensorFlow pour l'apprentissage automatique et ses différents atouts, tels que l'API de sous-classification et sa flexibilité.

Exemple d'utilisation

Enfin, si l'on veut retrouver le projet, il faut se rendre sur le lien github suivant : https://github.com/Mahadia/SNTP

Tout d'abord, il faut configurer sa machine en installant les logiciels pour veiller au bon fonctionnement du projet.

Création d'un projet

Exécuter **ng serve** pour un serveur de développement. Naviguez vers http://localhost:4200/. L'application sera automatiquement rechargée si l'un des fichiers source est modifié.

• Exécuter ng generate component nomDuComponent ou ng generate (directive |pipe|service|class|guard|interface|enum|module) pour générer un composant.

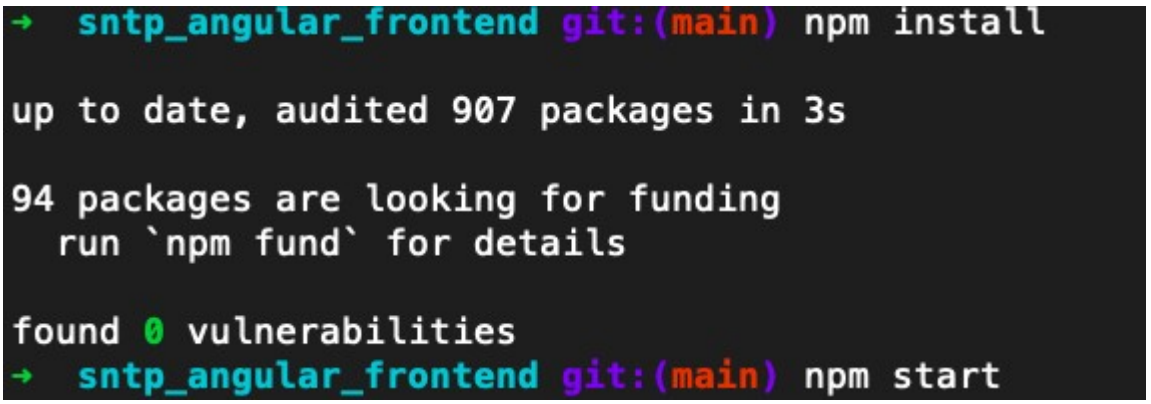
• Exécuter **ng build** pour construire le projet. Les artefacts de construction seront stockés dans le répertoire **dist/** .

• Exécuter **ng test** pour exécuter les tests unitaires via Karma.

• Exécuter **ng e2e** pour exécuter les tests de bout en bout via une plate-forme de votre choix. Pour utiliser cette commande vous devez d'abord ajouter un paquet qui implémente des capacités de test de bout en bout.

Lancement d'un projet existant

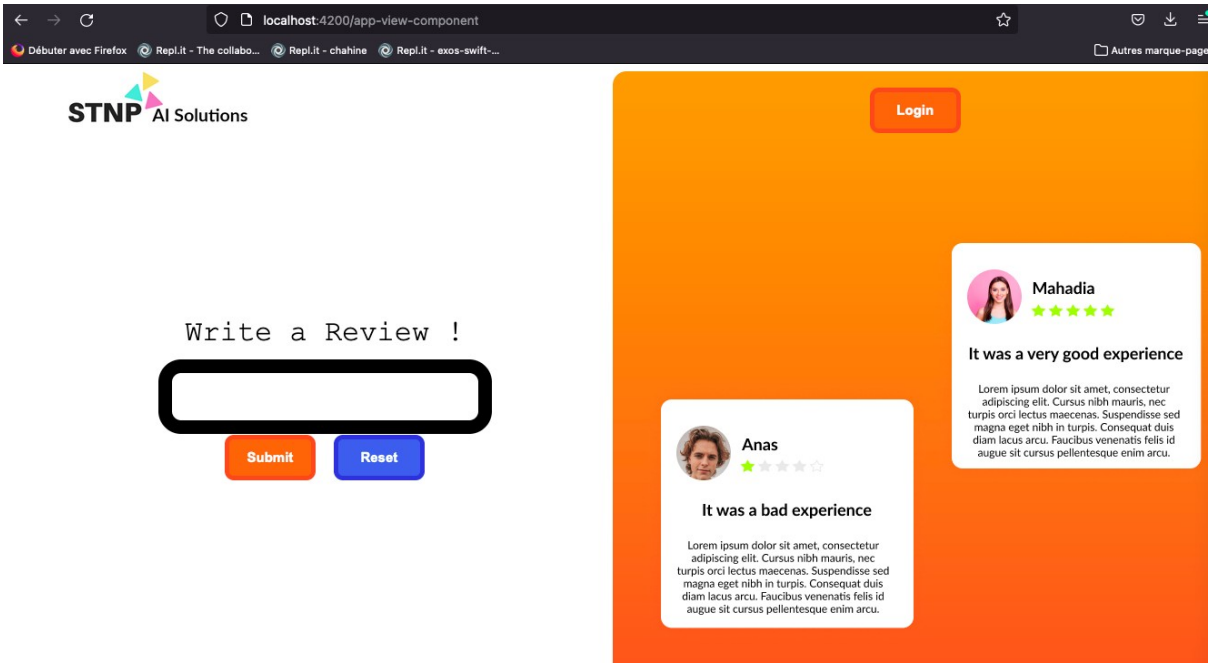
Dans un premier temps, via le terminal (Shell), se placer dans le dossier du projet SNTP puis exécuter la commande **npm install** qui permet d'installer toutes les dépendances pour l'application SNTP.



Ensuite pour lancer l'application, la commande **npm start** permet le lancement de l'application SNTP.

```
➜  sntp_angular_frontend git:(main) npm start

> ang-test@0.0.0 start
> ng serve

✔ Browser application bundle generation complete.

Initial Chunk Files   | Names         |  Raw Size
vendor.js             | vendor        |   3.17 MB |
polyfills.js          | polyfills     | 294.79 kB |
styles.css, styles.js | styles        | 251.23 kB |
main.js               | main          |  79.24 kB |
runtime.js            | runtime       |   6.51 kB |

                      | Initial Total |   3.79 MB

Build at: 2022-03-21T12:25:23.844Z - Hash: 1700cded4780e6f1 - Time: 31324ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **


✔ Compiled successfully.
```

Enfin, dans un moteur de recherche de notre choix, lancer l'adresse http://localhost:4200/ afin de visualiser l'interface utilisateur.



Faire une prédiction

Dans la zone de texte, l'utilisateur va entrer la phrase suivante : " I'm lovin it " .
Après avoir appuyé sur le bouton **Submit**, la réponse affichée sera positive, ce qui représente la prédiction, qui est bien en accord avec le sentiment du texte.

• POST : {action: "predict", text: "I'm lovin it"}
• RESPONSE : {prediction: "positive", text: "I'm lovin it"}

Enregistrer les données dans un ensemble de données

Si la réponse affichée est en accord avec le sentiment du texte, alors on devra appuyer sur le bouton **Yes** pour signifier que l'algorithme a détecté le type de commentaire et **No**, pour signifier que l'algorithme s'est trompé. Dans le cas de **"**I'm lovin it", le modèle a bien détecté que c'était un commentaire positif donc c'est un succès.

• POST : {action: "record", prediction: "positive", text: "I'm lovin it"}
• RESPONSE : {status: "success"}

Récupérer les données du moniteur

Nous savons que tous les 1 000 commentaires, une phase de réapprentissage est réalisée. Un nouveau modèle est alors créé et les performances des modèles précédents sont stockées et affichées.

• GET : {params: {target: "monitor_data"}}
• RESPONSE : {logs: [{ timestamp: "2022-03-11 12:28:23", winner: "new", old: {loss: 0.32,
accuracy: 0.87}, new: {loss: 0.29, accuracy: 0.89} }, ...] }

Récupérer le graphique d'entraînement le plus récent

On pourra observer deux images de graphes :

- l'accurancy
- loss

Ces deux graphes donnent les performances du modèle actuel.

• GET : {params: {target: "monitor_graph"}}
• RESPONSE : < PNG IMAGE >

Machine learning :

TF (TensorFlow) modèle Flask

L’image Architecture globale [4] ci-dessus, présente une nouvelle implémentation 'TF Inference Server' utilisant le micro-framework Flask.

- Pour démarrer le micro-framework Flask, il faudra lancer le fichier tf_model_server.py dans le répertoire 'tf_model_flask/tf_model_server.py'.
- Celui-ci fournit une API RESTful pour exécuter des prédictions sur le modèle d'IA, ajoute de nouveaux enregistrements à l'ensemble de données et récupère des informations sur le moniteur.
- Ensuite, après l'ajout de 1000 nouveaux échantillons de texte étiquetés à l'ensemble de données, le processus de renouvellement de l'IA se déclenche automatiquement.
- Enfin, le modèle nouvellement formé sera comparé à l'ancien modèle. Si le nouveau modèle est meilleur, l'ancien modèle est archivé et le nouveau modèle est mis en service.

Machine utilisée :

Processeur	Intel i7-7700k
Carte graphique	NVIDIA GTX1060
RAM	32 Go