T9 - Intelligence Artificielle

T-AIA-901

TrRésolution de commande avel éd

Traitement du langage naturel

1.1.6

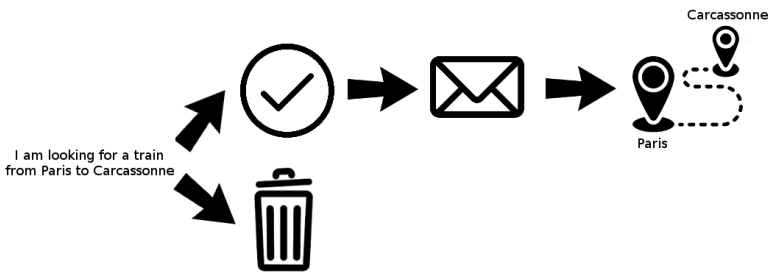
EPITECH.

Ordre de voyage résolu

méthode de livraison: GithubGenericName
nom du référentiel: $CourseCode-$GroupName.git
Langue: Python ou R ou toute autre chose qui fait le travail

Vous êtes censé créer un programme qui traite les commandes vocales et textuelles pour émettre un itinéraire approprié.

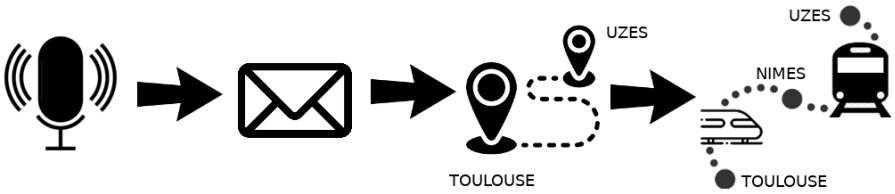
Plus précisément, votre logiciel recevra les commandes de voyage sous forme d'e-mail ou d'enregistrement téléphonique, et affichera au moins une ligne de voyage appropriée qui correspond aux attentes du client.



Vous ferez la distinction entre les commandes valides et non valides, et vous identifierez le départ et la destination.

De plus, vous pourriez avoir besoin d'étapes supplémentaires pour fournir une satisfaction absolue.

Avant la compréhension du texte, vous devez ajouter une étape de reconnaissance vocale, pour convertir les données vocales en données textuelles.



Après la compréhension du texte, vous pouvez ajouter un processus d'optimisation qui trouve le meilleur chemin dans un graphique de distance, afin d'obtenir les correspondances ferroviaires optimales.

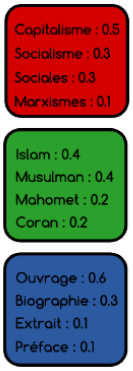
Nous supposons que vous maîtrisez déjà les bases de l'apprentissage automatique et que vous avez une certaine pratique avec les bibliothèques associées

OUTILS ET TECHNIQUES

TRAITEMENT DU LANGAGE NATUREL

Donc, vous voulez essentiellement extraire le sens d'un texte. La PNL est un domaine de recherche en constante expansion de plus de 50 ans, avec un large éventail de techniques. Cela signifie que vous avez beaucoup de recherches à effectuer et de choix à faire avant d'écrire une ligne de code.

Certains algorithmes fonctionnent avec de simples sacs de mots et utilisent des statistiques simples sur les données. Ils sont très utiles pour la catégorisation ou l'identification de l'auteur, du contexte, etc.



Certains autres algorithmes tentent de préserver l'ordre des mots ou de comprendre la grammaire de phrases spécifiques. Ils sont souvent nécessaires pour comprendre des informations fines sur de petites données.

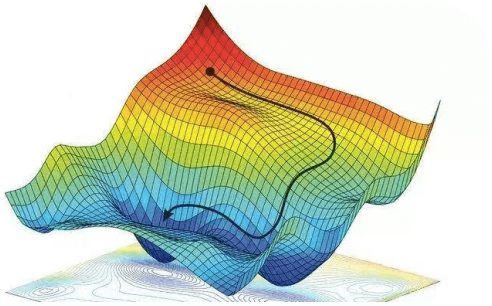
OPTIMISATION GRAPHIQUE

Trouver un chemin optimal dans un graphe est un problème déterministe, qui appartient à la**P**classer.

Évidemment, comme il y a un nombre fini de gares, une recherche par force brute fonctionnerait. Mais voulez-vous vraiment explorer un nombre exponentiel de routes ? Certainement pas vous !

Jetez un œil à la littérature sur l'algorithmique et la complexité, avant de vous lancer directement dans l'implémentation de votre algorithme. Il est souvent plus intéressant de le programmer soi-même à partir de rien.

Plus généralement, les algorithmes d'optimisation sont au cœur de nombreuses solutions de ML. Notamment le deep learning, qui est basé sur la rétropropagation.

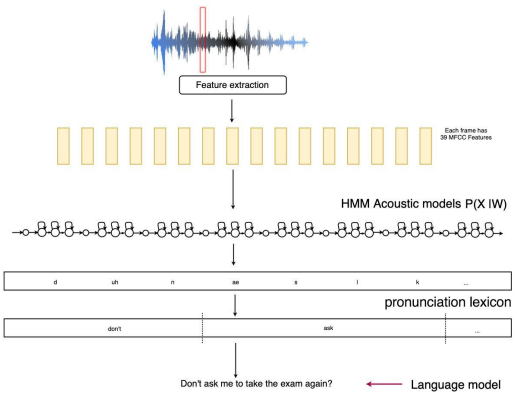


RECONNAISSANCE VOCALE

Puisqu'il s'agit d'un composant isolé de votre architecture et qu'il fait référence à un problème unique bien défini et bien étudié, la reconnaissance vocale devrait en fait être la partie la plus facile du travail ici.

Néanmoins, les mathématiques derrière la reconnaissance vocale sont extrêmement intéressantes, et nous conseillons vivement à ceux qui souhaitent se spécialiser dans la reconnaissance des signaux d'étudier le fonctionnement des modèles de Markov cachés.

Construire un tel modèle à partir de zéro dépasse de loin la portée de ce premier projet, mais il serait formidable de mieux comprendre comment les composants d'une boîte à outils de reconnaissance vocale fonctionnent ensemble.



CARACTÉRISTIQUES

ENTRÉE SORTIE

Le composant central (PNL) devrait être en mesure de :

- recevoir en entrée un fichier texte brut
- discriminer si le texte est un ordre de voyage**formulé en français**
    - retourner une réponse négative si ce n'est pas le cas
    - retourner une paire*Départ, Destination*Si c'est

Le composant vocal doit pouvoir :

- enregistrer un signal vocal
- renvoie un fichier texte si l'entrée est**formulé en français**

Le composant Pathfinder final devrait pouvoir :

- recevoir en entrée une paire*(Départ, Destination)*
- retourner un**trajet en train**, comme une séquence de villes*(Départ,Etape1,Etape2,...,Destination)*, ce qui est minime, en ce qui concerne le temps de trajet total parmi toutes les possibilités

JEUX DE DONNÉES

C'est votre travail de construire un ensemble de formation pour le composant PNL.

Veillez à inclure différents types de textes poubelles, et plus important encore, les commandes réelles explorent la variété des structures grammaticales

D'autre part, le jeu de données pour le pathfinder vous est donné au début du projet, vous pouvez donc évaluer les performances de votre algorithme au fur et à mesure que vous le construisez. C'est le dossier*horaires.csv*.

Par souci de simplicité, vous pouvez supposer que les modifications ne coûtent pas de temps supplémentaire (durée de A à C à B = durée de A à B + durée de B à C).

LIVRAISON

Votre livraison se compose du binaire (nous vous recommandons de séparer les trois composants afin qu'ils puissent être testés individuellement) et d'une documentation solide, qui comprend :

- l'architecture complète des différentes couches de votre application
- une description du processus de formation (y compris la livraison des ensembles) et les paramètres finaux
- un exemple détaillé de ce qui arrive à un texte donné tout au long du processus

PRIME

Vous pouvez améliorer ce projet de plusieurs manières, notamment :

- benchmarker différents modèles, selon des critères de performance
- répondre à d'autres types d'ordres que le voyage (manger, boire, dormir, ...)
- recherche par d'autres moyens de transport
- traitement d'autres langues que le français
- ajouter un protocole d'authentification vocale
- compte tenu des temps d'attente aux gares intermédiaires (*data_sncf.zip*pourrait vous aider avec ceci)

Faites attention dans ce dernier cas que le problème devienne beaucoup plus difficile de cette façon, à la fois pour l'analyse des données et à cause de la complexité des calculs.