

- + **ALE-SemanticSearch**
  - | **readme.md** (document expliquant tout ce qu'il y a à savoir concernant ce projet d'un point de vue utilisateur/développeur)
  - | **requirements.txt** (liste des bibliothèques python nécessaires à installer pour pouvoir exécuter les différents scripts python)
- +---**cache**
  - | ... (données cachées ici pour accélérer certains aspects de mes programmes)
- +---**Configs\_ConversationEngines**
  - | ... (fichiers de configurations pour mon moteur de découpe des conversations)
- +---**Configs\_NER\_Engines**
  - | ... (fichiers de configurations pour mon moteur de reconnaissance d'entités nommées)
- +---**Configs\_SearchEngines**
  - | ... (fichiers de configurations pour mon moteur de recherche)
- +---**DataToImport**
  - | ... (données de conversations sous le format d'export standard de Rainbow)
- +---**SavedImported\_RBI**
  - | ... (données de conversations importées sauvegardées sous un format simple à importer)
- +---**models**
  - ... (dossier où les différents modèles d'embedding ou de traduction utilisés seront téléchargés en local)
- +---**profiling\_results** (Dossier qui contient les données de profiling de mes scripts python + une page web pour pouvoir les visualiser / les analyser)
  - +---**results**
    - | ... (Dossier où sont sauvegardées les données de profiling des différentes exécutions de certains de mes programmes python)
  - +---**web\_visualisation** (Page web pour visualiser et analyser le temps d'exécution de différentes parties de certains de mes programmes python)
    - | **index.html**
  - +---**css**
    - | ... (styles css)
  - +---**js**
    - | ... (scripts js)
- +---**PythonScripts** (Dossier où il y a tous les différents programmes python de ce projet)
  - | **bubble.py** (Contient la classe Bulle définissant une bulle d'un environnement Rainbow)
  - | **config.json** (Fichier de configuration pour le serveur websocket python et les scripts sous-jacents, contient les configurations websocket ainsi que tous les chemins vers les données à sauvegarder ou à charger)
  - | **config.py** (Contient une classe représentant le fichier de configuration ci-dessus)
  - | **config\_socket\_api.json** (Fichier de configuration, mais pour le serveur socket auquel on va connecter le client C# qui va interagir avec l'API SDK Rainbow)
  - | **config\_socket\_api.py** (Contient une classe représentant le fichier de configuration ci-dessus)

- | **conversations\_algorithms.py** (Contient les différents algorithmes de découpe de conversations)
- | **conversations\_engine.py** (Contient le moteur de découpe de conversations)
- | **embeddings\_cache.py** (Contient un module qui va gérer les caches des embeddings des messages, pour ne pas avoir à les recalculer à chaque fois)
- | **embedding\_calculator.py** (Contient une interface générique pour calculer des embeddings depuis du texte)
- | **global\_variables.py** (Module simple à utiliser pour gérer des variables globales facilement et proprement dans un projet complexe en python)
- | **language\_translation.py** (Contient un module qui va gérer la traduction)
- | **lib.py** (Librairies de fonctions génériques non spéciales)
- | **lib\_date\_recognition.py** (Module de Reconnaissance de dates dans du texte basé sur des règles syntaxiques)
- | **lib\_embedding.py** (Contient la classe MessageEmbedding ainsi que quelques fonctions pour calculer des distances entre embeddings)
- | **lib\_hp\_optimization.py** (Contient des fonctions pour faire de l'optimisation d'hyper-paramètres)
- | **lib\_main\_server.py** (Contient la classe abstraite MainServer)
- | **lib\_number\_converter.py** (Contient un module de conversion de nombres écrits textuellement en format numérique, pour le script lib\_date\_recognition.py, ex: "Quatre cent vingt-trois" -> 423)
- | **lib\_types.py** (Contient une représentation des types et architectures de configurations pour l'optimisation des hyper-paramètres)
- | **main\_convert\_data\_to\_rbi.py** (Script principal à exécuter / Point d'entrée pour importer les données du format d'exportation de base de bulles rainbow en données structurées facilement utilisables pour ce projet)
- | **main\_server\_for\_api.py** (Script principal à exécuter / Point d'entrée pour le serveur pour le bot / l'API Rainbow)
- | **main\_server\_for\_webapp.py** (Script principal à exécuter / Point d'entrée pour la webapp démo et l'optimisation des hyper-paramètres)
- | **main\_tests\_benchmarks.py** (Script principal à exécuter / Point d'entrée pour lancer les benchmarks sur les différentes configurations de mes moteurs (recherche, conversations et NER))
- | **message.py** (Contient la classe Message définissant un message d'un environnement Rainbow)
- | **ner\_algorithms.py** (Contient les différents algorithmes de reconnaissance d'entités nommées)
- | **ner\_engine.py** (Contient le moteur de reconnaissance d'entités nommées)
- | **profiling.py** (Contient un module pour faire du profiling de code et voir / analyser facilement le temps d'exécution de mes différents portions de codes python -> affichage des résultats via une page web statique dans le dossier profiling\_results)
- | **rainbow\_instance.py** (Contient la classe RainbowInstance définissant une instance Rainbow (RBI) / un environnement Rainbow)
- | **search\_algorithm.py** (Contient les différents algorithmes de recherche)
- | **search\_engine.py** (Contient le moteur de recherche)

- | **threads\_tasks\_server.py** (Contient le serveur multi-tâches qui va exécuter sur plusieurs threads les tâches demandées avec un système de file d'attente)
- | **user.py** (Contient la classe User définissant un utilisateur d'un environnement Rainbow)
- +---**Rainbow\_App** (Dossier contenant la partie C#)
  - | **api\_config.json** (Fichier de configuration pour l'application C#)
  - | **MainRainbowAppSDKApi.cs** (Programme principal C# client socket au serveur python et utilisant l'API SDK Rainbow)
- +---**TestsBenchmarks**
  - | ... (Fichiers de benchmarks pour les différentes tâches: search, NER et découpe de conversation)
- +---**WebApp** (Dossier contenant la WebApp principale de mon projet, avec une démo, des benchmarks, et la page d'optimisation des hyper-paramètres. Les pages de la démo et de l'optimisation des hyper-paramètres nécessitent que le serveur python des webapps soit lancé)
  - | **benchmarks.html** (page d'accueil des benchmarks, redirigeant vers les trois différentes pages de benchmark juste ci-dessous)
  - | **benchmarks\_auto\_ner.html** (Page de benchmark de la tâche de NER)
  - | **benchmarks\_conversation\_cutting.html** (page de benchmark de la tâche de découpe des conversations)
  - | **benchmarks\_semantic\_search.html** (page de benchmark de la tâche de recherche)
  - | **demo.html** (page web de la démo pour la recherche)
  - | **explications.html** (page web qui contient des explications générales sur comment fonctionne ce projet)
  - | **hyper\_parameters\_optimisation.html** (page web qui contient l'application d'optimisation des hyper-paramètres)
  - | **index.html** (page d'accueil redirigeant vers les benchmarks, la démo, ou bien les explications)
- +---**css** (Dossier avec les styles CSS)
  - | ...
- +---**data\_benchmarks** (Dossier où les scripts python sauvegardes les résultats des benchmarks)
  - | **benchmark\_results.js** (Script js contenant les données des deux fichiers json ci-dessous dans des variables, pour pouvoir y accéder facilement depuis les autres scripts js, fichier généré dynamiquement par les scripts python)
  - | **benchmark\_results.json** (Données brutes json contenant les résultats des benchmarks testés)
  - | **data\_benchmarks.json** (Données brutes json contenant les infos sur les benchmarks testés)
- +---**js** (Dossier avec les scripts js)
  - | ...