Projet TAL - RI

L'objectif du projet est de réaliser un système de recherche d'information dans une collection de descriptions de films publiées sur Allociné. Le projet se décompose en 2 étapes (voir ci-dessous pour le détail).

Le projet se fera par groupe de **2 étudiant·e·s** et le rendu final inclura :

- le code source (en Python, sous forme de notebooks Jupyter)
- les données utilisées (ressources externes éventuelles, corpus, modèles, etc.)
- un rapport de 6 pages qui expliquera votre méthode, les expériences réalisées et les scores obtenus, la répartition du travail au sein du groupe, une bibliographie (références / sites consultés pour réaliser le travail)

Précisions sur les fichiers à rendre pour l'étape 2 du projet : Déposez le core Solr complet dans un zip et le(s) fichier(s) contenant les descriptions de films que vous avez indexées.

Les fichiers pourront être déposés sur Moodle ou sur git.unistra.fr.

Date limite pour rendre le projet : **30 avril 2023**

Toutes les questions sur le projet devront être posées sur le forum dédié sur Moodle : https://moodle.unistra.fr/mod/forum/view.php?id=131934

Etape 1

Vous avez à votre disposition sur Moodle deux fichiers CSV (`allocine_genres_train.csv` et `allocine_genres_test.csv`) de descriptions de films rédigées en français ; le contenu provient d'Allociné. Le titre de chaque film se trouve dans la colonne 'titre', une synopsis dans la colonne 'synopsis' et le genre du film dans la colonne 'genre', qui est la dernière. D'autres informations sont également disponibles pour chaque film, comme les réalisateur·trice·s, quelques acteur·trice·s, année de sortie, langue et nationalité etc.

L'objectif est d'entraîner un outil de classification automatique des films en fonction de leur genre. La classification doit se baser sur le texte de la synopsis et sur le titre des films. Le texte et le titre des articles ont déjà été tokénisés et tous les tokens sont séparés par une espace.

Les différents tests de comparaisons entre pré-traitements et modèles, ainsi que l'entraînement, se feront à partir du fichier `allocine_genres_train.csv` et à l'aide des méthodes vues lors des travaux pratiques. Vous êtes fortement encouragés à comparer différents algorithmes de classification et différents types de traits déduits du texte des articles (par exemple, utilisation de la désuffixation, application de différents seuils de fréquence pour les tokens à conserver dans le vocabulaire, utilisation de différentes méthodes de pondération, etc. : à vous de trouver et de tester des traits éventuellement pertinents). Les bonnes pratiques d'évaluation des modèles (validation croisée notamment) devront être respectées. L'analyse des résultats comportera obligatoirement une analyse qualitative (comment expliquer les problèmes observés ?) et pistes d'amélioration.

En fonction des résultats de vos expériences, vous sélectionnerez et entraînerez le modèle final à partir du jeu de données `allocine_genres_train.csv`. Ce modèle sera ensuite appliqué à `allocine_genres_test.csv` pour effectuer des prédictions. La colonne genre ne pourra pas être utilisée pour effectuer ces prédictions et vous devrez procéder comme si elle était absente. Elle sera toutefois conservée dans les données étiquetées produites, avec le genre prédit (cf. partie suivante).

Pour appliquer un modèle à un nouveau jeu de données, vous pouvez vous utiliser les méthodes suivantes :

- Pour scikit-learn : utiliser la méthode predict()
- Pour Keras : utiliser la méthode predict() et np.argmax pour trouver la classe avec la probabilité la plus élevée (cf. https://keras.io/api/models/model_training_apis/)
- Pour les transformers HuggingFace : utiliser la méthode predict() et np.argmax pour trouver la classe prédite

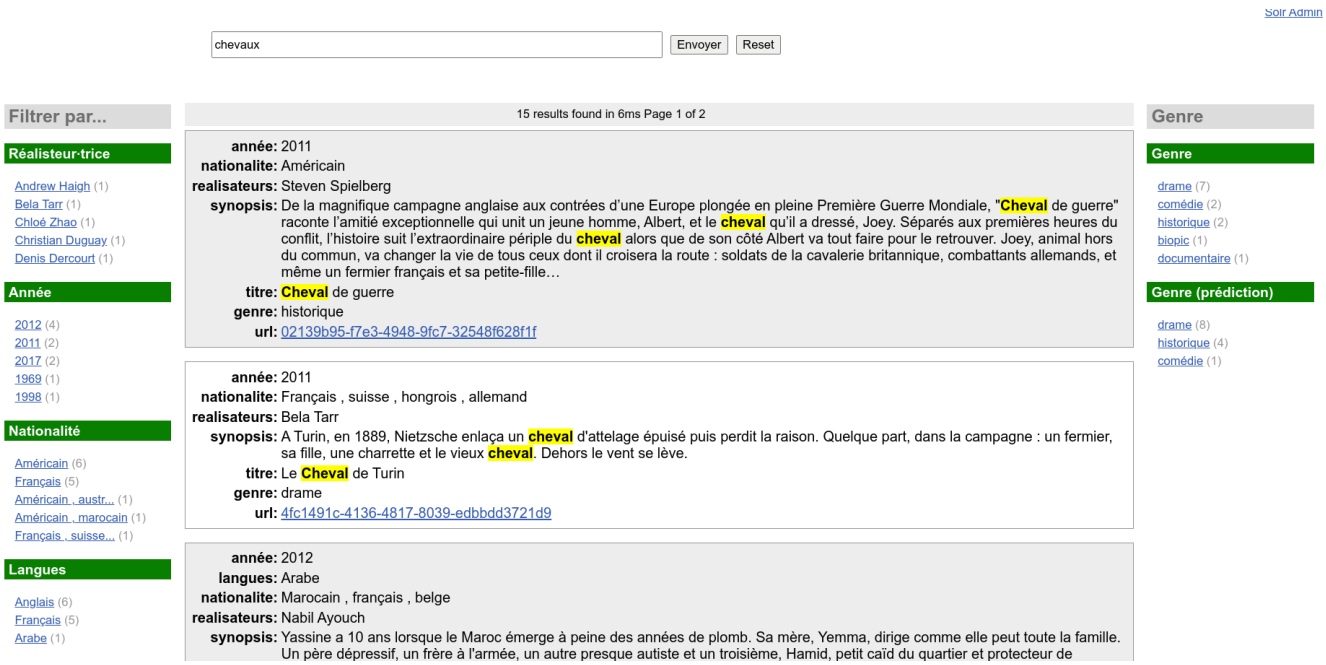
Etape 2

Le corpus `allocine_genres_test.csv` enrichi du genre prédit sera indexé dans une instance du serveur de recherche Solr[1]. Vous allez ensuite créer une interface simple de recherche en texte intégral, qui fournira également la possibilité de filtrer les films selon les caractéristiques disponibles dans le jeu de données, comme la langue, nationalité, réalisateur·trice, année ou autres. La fonction de recherche par facettes de Solr sera exploitée pour ce filtrage. L'image dessous donne un exemple de ce qui est attendu.

- Dans le panneau de gauche, donnez entre deux et quatre facettes avec des informations pour filtrer le film. Dans l'exemple (image ci-dessous), l'année de production, le réalisateur ou réalisatrice, la nationalité et la langue du film on été affichées comme facettes.
- Nous allons profiter du panneau de droite pour les informations suivantes : Le genre du film sera affiché, tant le genre déjà disponible dans le corpus test utilisé à l'étape 1 (facette « Genre ») que le genre prédit par votre outil de classification automatique (facette « Genre (prédiction) ». Avant l'indexation dans Solr, ajoutez une colonne au corpus CSV avec le genre prédit, et affichez tant le genre original que le genre prédit sur l'interface.

S'agissant d'un corpus en français, vous devrez utiliser la configuration linguistique de Solr adaptée pour le français. Dans l'exemple ci-dessous, on voit comment chercher le mot « chevaux » (au pluriel) récupère des documents qui contiennent le singulier (« cheval ») autant que le pluriel.

1 https://solr.apache.org/ La version [8.3] sera utilisée. L'outil aura été présenté lors des séances de cours. L'outil permet d'indexer des fichiers CSV directement.



Exemple de l'interface qui peut être créée. Dans cet exemple, la configuration Solr et les templates du TP ont été adaptés pour afficher quatre facettes à gauche et à droite deux facettes pour le genre (celui disponible dans les données et celui prédit à l'étape 1 du projet)