

CC_ratrapage

June 11, 2021

1 Examen de rattrapage Approches Cognitives

2020-2021, Master LLCER - parcours TAL

Iana Atanassova, Nicolas Gutehrlé

1.0.1 Modalités

Date et durée de l'examen : Lundi 14 juin, de 8h à 10h, en ligne sur Teams à :

<https://teams.microsoft.com/l/team/19%3a653c450830464ec9a388d47f1dd551c9%40thread.tacv2/conversations?groupId=6f60-42ae-8b5b-d5105bab7b64&tenantId=ca5a017f-00e6-423c-a8c6-0a1b459f6827>

L'utilisation d'Internet ainsi que des supports de cours est **autorisée**. Le travail est **strictement personnel**. L'évaluation portera sur la compréhension des notions du cours qui sera démontrée par l'étudiant. Un entretien oral est possible suite à l'examen pour préciser la note.

Vous répondrez à au moins 3 des quatre exercices ci-dessous.

Les copies devront être rendues par mail à 10h au plus tard à : * nicolas.gutehrlé@univ-fcomte.fr
* iana.atanassova@univ-fcomte.fr

Un point sera retiré pour toutes 5 minutes de retard.

1.1 Exercice 1 (20 min)

Expliquez les différences qui existent entre les approches par règles linguistiques (grammaires, dictionnaires, etc.) et les approches par apprentissage (machine learning et deep learning).

Quels sont les avantages et les inconvénients dans les deux cas ?

1/2 page suffit (1 ou 2 paragraphes).

1.2 Exercice 2 (20 min)

Un algorithme KNN est entraîné pour distinguer les verbes et les noms en français. Il utilise la distance D qui est définie comme suit :

$D(m_1, m_2)$ entre deux mots m_1 et m_2 est le nombre de différences qui existent entre les 2 mots en considérant leurs 3 premiers caractères et leurs 3 derniers caractères.

Par exemple : - pour calculer la distance entre “aller” et “parler”, nous comparons les débuts des mots “all” avec “par” (3 différences, pour le 1e, 2e et 3e caractère) et les fins “ler” avec “ler” (0 différences). Donc $D(aller, parler) = 3$. - pour calculer la distance entre “terre” et “mettre”,

nous comparons “ter” avec “met” (2 différences, pour le 1e et 3e caractères) et “rre” avec “tre” (1 différence pour le 1e caractère). Donc $D(terre, mettre) = 3$.

Le corpus d’apprentissage est le suivant :

- aller - v.
- venir - v.
- porter - v.
- terminer - v.
- promettre - v.
- pallier - n.
- porche - n.
- fenêtre - n.
- finale - n.

Quels seront les réponses de l’algorithme pour les mots : “vernir” et “traître” pour $k = 3$? Expliquez pourquoi.

Est-ce que cet algorithme est bien adapté pour la tâche d’analyse morphologique des noms et verbes ? Expliquez pourquoi.

1.3 Exercice 3 (20 min)

Créez un programme qui, à l’aide de la librairie `pandas`, ouvre le fichier `multilingual_reviews.csv`, et affiche les informations suivantes: * les informations à propos du fichier : combien de lignes le fichier comporte-t-il ? * les 5 premières lignes du fichier. * toutes les valeurs distinctes de la colonne “product_category” ainsi que le nombre d’éléments qui contiennent chacune de ces valeurs

1.4 Exercice 4 (60 min)

Créez un programme qui, à l’aide de la librairie `pandas`, ouvre le fichier `wine.csv` contenu dans le dossier `data`. Ce fichier contient 14 colonnes: les 13 premières décrivent différentes propriétés de vins. La dernière colonne correspond à la catégorie du vin.

A partir de ce fichier, vous entraînerez **un modèle de classification KNN** à l’aide de la classe `KNeighborsClassifier` de `scikit-learn`. Ce modèle doit apprendre à prédire la classe d’un vin à partir des 13 premières colonnes.

Vous pourrez diviser les données à l’aide de la fonction `train_test_split` pour le train et test set.

Vous entraînerez le modèle sur le train set et validerez l’entraînement sur le test set. Vous utiliserez la fonction `score` pour afficher les performances du modèles.