DEFT 2022 – Guide de participation

Tâche continue

1. Présentation

Contrairement à la **tâche de base** qui consiste à construire un système de prédictions de notes d'étudiants (entre 0 et 1) à partir d'une référence (construction classique d'un système ou modèle sur le corpus d'entraînement et application sur le corpus de test), la **tâche continue** permet aux participants d'améliorer en permanence leur système en obtenant les notes réelles obtenues par les étudiants. Du point de vue de l'enseignant, il s'agit de fixer un pourcentage de corrections acceptable. Par exemple, si le taux d'erreur estimé est de 10 %, les apprenants pourront contester mais cela restera un nombre gérable pour l'enseignant, tout en motivant les apprenants à chercher s'il n'y a pas d'erreur de corrections pour leur copie. Pour que le système considère être en dessous de ce seuil d'erreur, il choisira le plus petits nombre de copies pertinentes.

Dans cette perspective, l'objectif est d'élaborer une *stratégie d'interrogation* pertinente du serveur *(exploration dynamique d'espace)* pour améliorer son système en continue. La stratégie de base consiste à identifier les réponses identiques entre étudiants, à demander la note d'un seul étudiant, et à attribuer cette même note aux étudiants ayant produit une réponse identique, puis à faire de même sur un deuxième ensemble de réponses, etc.

Ce guide présente le déroulement d'une participation à la tâche de prédictions de notes en continue.

2. Organisation

Les notes obtenues par chaque étudiant à chacune des questions des corpus d'entraînement et de test sont stockées sur un serveur distant. Un cycle d'interrogation et d'amélioration du système passe par trois étapes :

1. le participant demande de la note d'un étudiant pour une question : le serveur renvoie la note demandée
2. le participant adapte son système en conséquence et prédit les notes de tous les étudiants sur cette question
3. le participant soumet cet ensemble de prédiction sur le serveur

Tant que ce cycle n'est pas achevé, il est impossible de demander une nouvelle note. Il n'est pas non plus possible de demander de nouveau la note d'un étudiant, d'où l'intérêt d'élaborer une stratégie efficace.

Le même serveur est utilisé pour *obtenir la référence* et pour *soumettre ses prédictions*.

3. Scripts

Pour faciliter ce processus itératif, quatre scripts python ont été produits et sont mis à disposition des participants. Les trois premiers scripts renvoient aux trois étapes précédentes, le quatrième script permet de vider la base de données.

Chaque participant dispose de sa propre base de données sur notre serveur. Nous lui communiquerons les informations de connexion ultérieurement (y compris l'adresse du serveur) :

- son nom d'utilisateur LOGIN
- son mot de passe PASS
- le nom de sa base de données BDD

3.1. Installation

Les scripts ont été conçus et testés sous une distribution Ubuntu 20. Ils nécessitent l'installation des modules suivants :

- sudo apt-get install python3-dev libmysqlclient-dev
- pip install mysqlclient mysql-connector-python
- pip install mysqlclient sqlalchemy

Chaque script appelle des fonctions prédéfinies sur le serveur.

3.2. Détail

Les portions sur fond jaune sont celles à modifier à chaque itération (typiquement, le numéro de la question et l'identifiant de l'étudiant pour qui la note est demandée).

Script 01_ddeNote.py

Ce script permet de demander la note d'un étudiant à une question. Dans cet exemple, nous demandons la note obtenue par l'étudiant student116 à la question 1002.

```
import mysql.connector
try:
	mydb = mysql.connector.connect(
		host="SERVEUR",
		user="LOGIN",
		password="PASS",
		database="BDD",
		autocommit=True,
		ssl_disabled= True
	)
	mycursor = mydb.cursor()
	mycursor.callproc('GetCorrection', ['1002','student116'])
	for i in mycursor.stored_results(): results = i.fetchall()
	print (results[0][0])
	mystr= 'note recup ' + str(results[0][0])
	mydb.close()
except Exception as e:
	print ("Exception\n", e)
```

L'information retournée est la note, par exemple : « 1.0 »

A l'issue de cette interrogation, le participant génère par ses propres moyens un fichier contenant les prédictions de notes pour cette question, composé de cinq colonnes (numéro de la question, identifiant de l'étudiant pour qui la note a été demandée, identifiant de l'étudiant pour qui la prédiction est faite, note prédite, confiance du système dans la note), avec la tabulation comme séparateur :

QNUM	e_dem etudiant		note	Confiance
1002	student116	student101	1	0.5
1002	student116	student108	0	0.5
1002	student116	student116	1	0.5
1002	student116	student121	1	0.5
1002	student116	student122	1	0.5
1002	student116	student13	1	0.5
1002	student116	student3	1	0.5
1002	student116	student46	0	0.5

N.B. : si votre système ne calcule pas de score de confiance, mettre toujours « 0 ».

Script 02_sauveNotes.py

Ce script permet de charger le fichier de prédictions précédemment réalisé (1002_s116.csv) et de le déposer sur le serveur distant. Le fichier est nommé avec le numéro de la question (1002) et l'identifiant de l'étudiant (s116) pour lequel la note vient d'être demandée (convention de nommage facultative).

```
import csv
import pandas as pd

from sqlalchemy import create_engine
try:
        engine = create_engine('mysql://LOGIN:PASS@SERVEUR/BDD?ssl_mode=DISABLED')
        df = pd.read_csv("./1002_s116.csv",sep='\t',quotechar='\'',encoding='utf8')
        df.to_sql('PROPOSITION',con=engine,index=False,if_exists='append')

except Exception as  e:
        print ("Exception\n", e)
```

Si aucune erreur ne s'est produite, rien n'est affiché à l'écran.

Script 03_soumetRun.py

Ce script permet de soumettre le run, identifié par le numéro de question (1002) et l'identifiant de l'étudiant pour lequel la note a été demandée (student116). Le participant peut laisser un commentaire qui sera enregistré dans la base de données.

```
import mysql.connector
try:
        mydb = mysql.connector.connect(
                host="SERVEUR",
                user="LOGIN",
                password="PASS",
                database="BDD",
                autocommit=True,
                ssl_disabled= True
        )
        mycursor = mydb.cursor()
        commentaire = 'Voici le commentaire 116'
        mycursor.callproc('SubmitRun', ['1002','student116',commentaire])
        mydb.close()
except Exception as  e:
        print ("Exception\n", e)
```

Si aucune erreur ne s'est produite, rien n'est affiché à l'écran.

Script 04_supAllRun.py

Ce dernier script permet de supprimer tous les runs de la base de données du participant. Le script trouve son utilité lors de la phase d'entraînement, d'abord pour prendre en main les scripts, puis pour tester une nouvelle stratégie.

```
import mysql.connector
try:
	mydb = mysql.connector.connect(
		host="SERVEUR",
		user="LOGIN",
		password="PASS",
		database="BDD",
		autocommit=True,
		ssl_disabled= True
	)
	mycursor = mydb.cursor()
	commentaire = 'Voici le commentaire'
	mycursor.callproc('EffaceRun')
	mydb.close()
except Exception as  e:
	print ("Exception\n", e)
```

Si aucune erreur ne s'est produite, rien n'est affiché à l'écran.