Centrale Méditerranée

Proposition de sujet d'alternance 1A
2023-24

Laboratoire : LIS

Titre du sujet : Analyse des trajectoires d'étudiants lors de l'apprentissage de la programmation

Encadrant * (s) : F. Flouvat (LIS), N. Durand (LIS) et M. Quafafou (LIS)

Nom :	Flouvat
Prénom :	Frédéric
Qualité ** :	Maître de conférences HdR
Localisation :	IUT d'Aix-en-Provence et bâtiment Polytech GII, Campus de St Jérôme, Marseille
Coordonnées *(e-mail/tel)*	frederic.flouvat@univ-amu.fr, nicolas.durand@univ-amu.fr, mohamed.quafafou@univ-amu.fr

* *un co-encadrement est possible.*

** *l'encadrement devra être assuré de préférence par un permanent du laboratoire, au* ***minimum titulaire d'un Doctorat****.*

Descriptif du sujet et de la mission (au moins sur la 1er année) :

L'apprentissage de la programmation passe de plus en plus par l'utilisation de plates-formes d'entraînement en ligne. Classiquement, les apprenants y soumettent leurs codes et la plate-forme leur retourne les éventuelles erreurs syntaxiques ou fonctionnelles sur la base de cas de tests définis par l'enseignant. L'exploitation des données de ces plates-formes ouvre des perspectives exaltantes en matière de suivi et d'aide à l'apprentissage de la programmation. Pour cela, des méthodes de fouille de textes sont notamment utilisées ces dernières années pour analyser ce type de données. Certaines approches essaient par exemple d'apprendre des représentations du code sous forme de vecteurs de réels, encore appelés « embeddings » (Q. Le and T. Mikolov 2014). Ces représentations permettent de projeter tout un vocabulaire dans un espace vectoriel de faible dimension, tout en capturant des aspects sémantiques. Avoir une telle représentation du code permet ensuite d'exploiter une grande diversité de méthodes existantes en intelligence artificielle (extraction de motifs, réseaux de neurones, SVM, clustering, etc.).

L'objectif de ce stage est d'analyser les trajectoires (de réussites ou d'échecs) suivies par les étudiants pour résoudre les exercices (car les étudiants peuvent faire de multiples tentatives), le tout en s'appuyant sur des « embeddings ». Dans un premier temps, il faudra faire une étude bibliographique des travaux visant à analyser des programmes, notamment dans un contexte d'éducation. On s'intéressera plus particulièrement aux méthodes permettant de faire de l'apprentissage d'« embeddings », ainsi qu'aux approches permettant d'exploiter ces représentations vectorielles pour analyser les « trajectoires » suivies par les étudiants pour concevoir leurs programmes. Une fois le travail bibliographique effectué, il faudra sélectionner certaines de ces méthodes et les tester sur un jeu de données réels composé de plus de 5000 programmes Python collectés dans le cadre d'un cours de première année de Licence. Une

interface de visualisation devra être construite afin de mettre en avant l'évolution dans le temps des programmes. Pour cela, un algorithme de réduction de dimensions tel que TSNE pourra aussi être utilisé.

Quelques Références :

U. Alon, M. Zilberstein, O. Levy, and E. Yahav. code2vec: Learning distributed representations of code. Proceedings of the ACM on Programming Languages, 3(POPL):1–29, 2019.

R. Bazzocchi, M. Flemming, and L. Zhang. Analyzing cs1 student code using code embeddings. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education, pages 1293–1293, 2020.

Q. Le and T. Mikolov. Distributed representations of sentences and documents. In International conference on machine learning, pages 1188–1196, 2014.

M. LJPvd and G. Hinton. Visualizing high-dimensional data using t-sne. J Mach Learn Res, 9:2579–2605, 2008.

Validation pour mise en ligne ECM :