

Apprentissage d'embeddings de codes pour l'enseignement de la programmation : une approche fondée sur l'analyse des traces d'exécution

Guillaume Cleuziou^{*,**}, Frédéric Flouvat^{*}

^{*} ISEA, Université de la Nouvelle-Calédonie, BP R4, 98851 Nouméa, Nouvelle-Calédonie
prenom.nom@unc.nc

^{**} Université d'Orléans, INSA Centre Val de Loire, LIFO EA 4022, Orléans, France
prenom.nom@univ-orleans.fr

Résumé. Améliorer l'efficacité pédagogique des plateformes d'entraînement à la programmation est une problématique en pleine effervescence qui nécessite de construire des représentations fines et exploitables des programmes d'apprenants. Cet article présente une nouvelle approche pour l'apprentissage d'embeddings de programmes. Partant de l'hypothèse que la fonctionnalité d'un programme, mais aussi son "style", peuvent être capturés par l'analyse des traces d'exécutions, la méthode *code2aes2vec* procède en deux étapes. Une première étape génère des séquences d'exécutions abstraites (AES) à partir de tests unitaires et des arbres syntaxiques abstraits (AST) des programmes soumis. La méthode *doc2vec* est ensuite utilisée pour apprendre des représentations vectorielles condensées (embeddings) des programmes à partir de ces AES. Cette contribution donne également lieu à l'exploitation et la mise à disposition de nouveaux jeux de données réelles. Une première évaluation réalisée sur ces données montre que les embeddings générés par *code2aes2vec* semblent capturer efficacement la fonctionnalité et le style des programmes.

1 Introduction

L'apprentissage de la programmation passe de plus en plus par l'utilisation de plateformes d'entraînement en ligne. Classiquement, les apprenants y soumettent leur(s) code(s) 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 excitantes en matière de suivi et d'aide à l'apprentissage de la programmation. Elles peuvent être utilisées par exemple pour identifier des étudiants en situation de décrochage, pour cibler des mauvaises pratiques ou pour propager des retours de l'enseignant. Ces fonctionnalités permettraient d'offrir à l'apprenant plus d'autonomie dans son apprentissage, et à l'enseignant d'être plus réactif et efficace dans ses interventions. Toutefois, cette exploitation nécessite une analyse fine des programmes soumis. Ces plates-formes d'entraînement doivent aller au-delà d'une simple analyse syntaxique du script, et permettre de considérer la sémantique associée.