Learning student program embeddings using abstract execution traces

Auteurs : Guillaume Cleuziou, Frédéric Flouvat

Présentateur : Hao ZHANG

8 juillet 2021

Sommaire

Contexte de recherche

- Généralement, dans les plateformes d'apprentissage de programmation, les apprenants soumettent leur(s) code(s) et la plateforme renvoie toute erreur de syntaxe ou erreur fonctionnelle. L'exploitation de ces données pourra ouvrir de nouvelles perspectives pour suivre et aider les débutants pour apprendre la programmation.
- Cependant, ces plateformes de formation doivent aller au-delà d'une simple analyse syntaxique du script, et permettre de considérer la sémantique associée. À cette fin, l'intégration de programmes d'apprentissage est récemment apparue comme un domaine de recherche prometteur.
- Inspirés du Text Mining, les programmes informatiques peuvent être traités comme du texte et utilisés pour générer une représentation vectorielle et compressée.

Le défi - I

Au niveau des caractéristiques du code

- Le code a certaines spécificités qui doivent être intégrées pour avoir des représentations aussi riches. Contrairement aux textes, les codes sont exécutables et de petites modifications peuvent avoir des impacts significatifs sur leurs exécutions.
- Un programme peut également appeler d'autres programmes qui peuvent eux-mêmes appeler d'autres programmes. Le contexte dans lequel une instruction est utilisée est également particulièrement important pour en déduire son rôle.
- Enfin, contrairement aux textes, les arbres syntaxiques des programmes sont généralement plus profonds et composés de sous-structures répétitives (boucles).

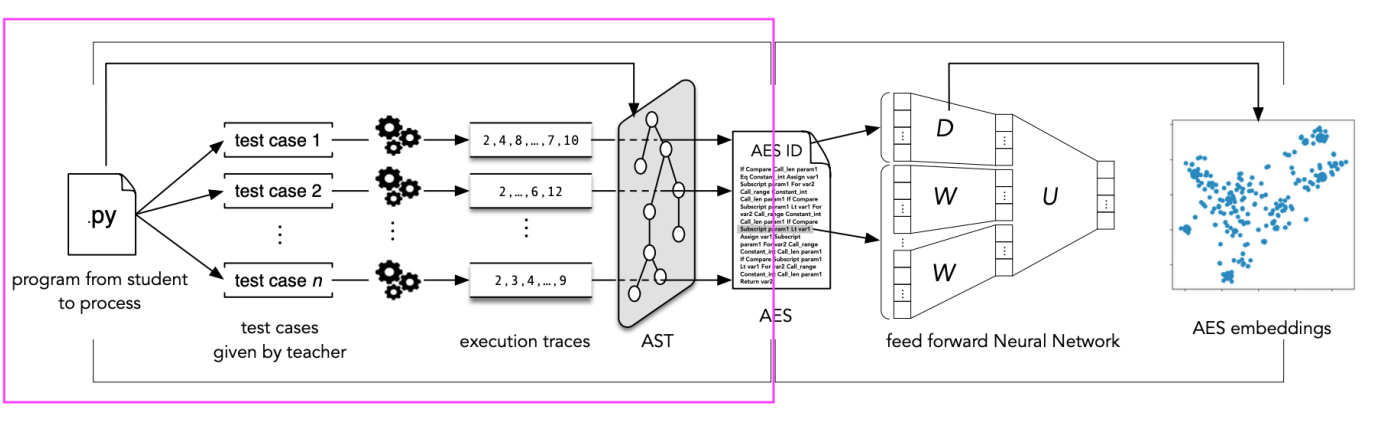
Le défi - II

Au niveau des approches existantes

- Les méthodes existantes pour la construction de l'intégration de programmes intègrent partiellement ces caractéristiques.
- Les approches existantes se concentrent davantage sur la fonction du programme que sur son style.
- De plus, la plupart de ces approches sont supervisées et construisent des intégrations pour une tâche spécifique.

Sommaire

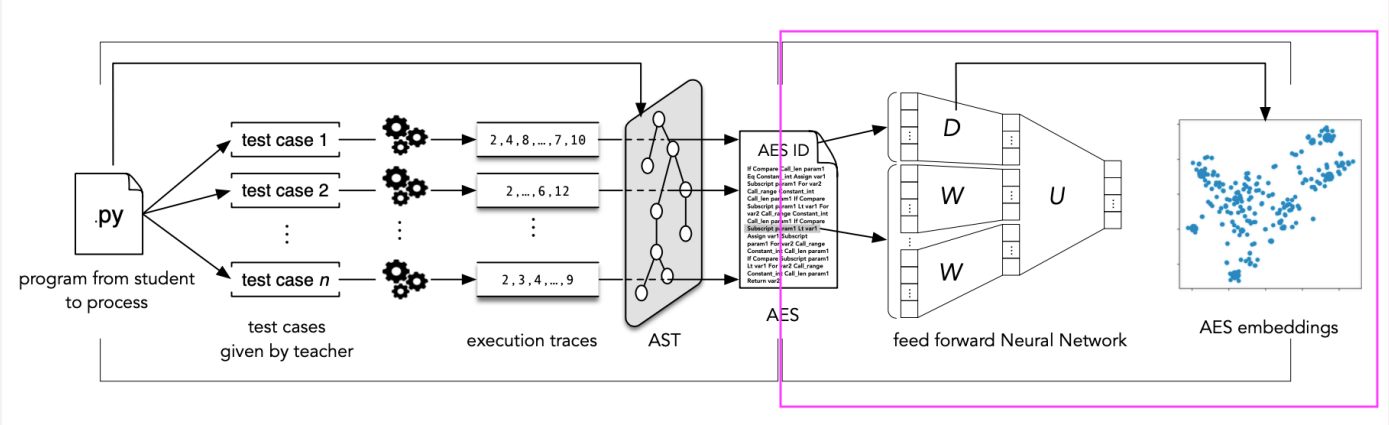
Deux étapes - I



Étape *code2aes*

Cette étape représente un programme sous la forme d'une séquence d'exécution abstraite (AES), correspondant aux chemins d'arbres de syntaxe abstraite (AST) utilisés par le programme lors de son exécution sur des cas de test prédéfinis.

Deux étapes - II



Étape *aes2vec*

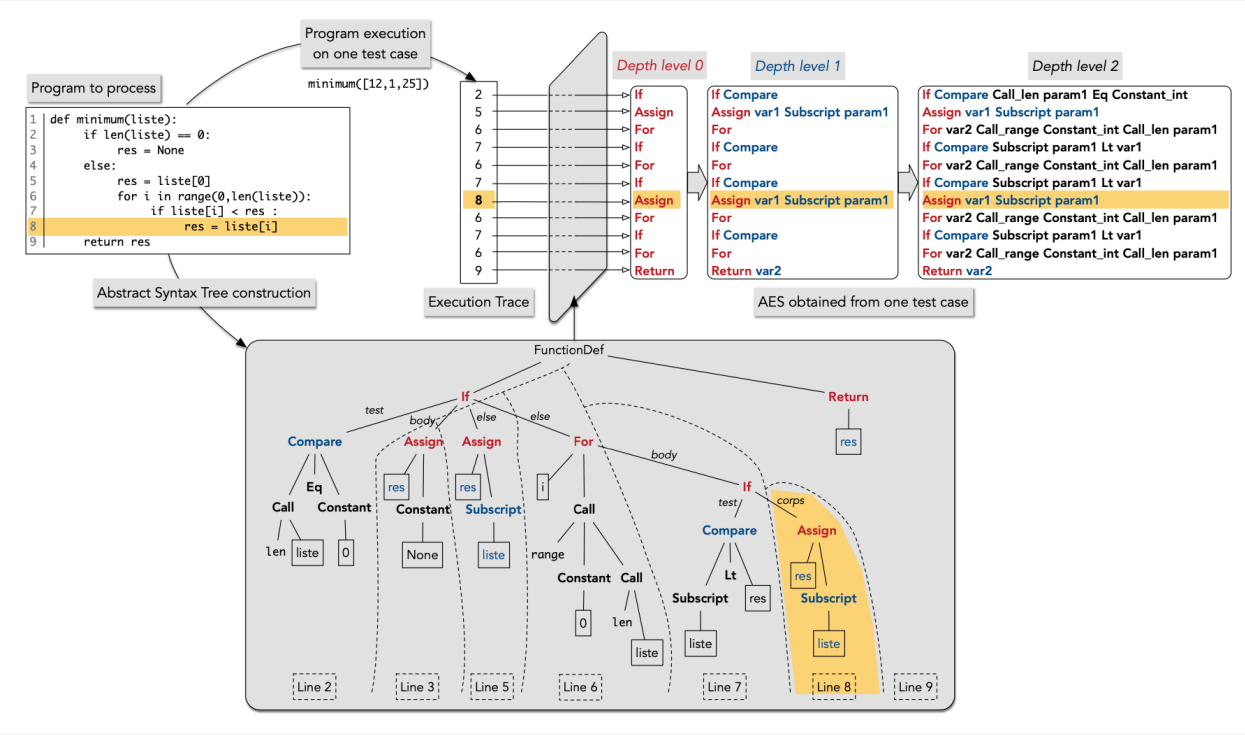
Cette étape utilise un réseau de neurones pour construire de l'intégration de programmes en fonction de leur AES (en utilisant l'approche *doc2vec*).

code2aes : construction de séquences d'exécution abstraites (AES) - I

Trois niveaux de traduction (ou d'abstraction) sont proposés selon la profondeur considérée dans l'AST :

- AES niveau 0 : chaque ligne de programme est représentée par un mot unique correspondant au symbole de tête du sous-arbre associé dans l'AST (en rouge sur la figure)
- AES niveau 1 : chaque ligne de programme est représentée par un ou plusieurs mots correspondant aux symboles de tête du sous-arbre associé et de ses sous-arbres principaux (en rouge et bleu sur la figure),
- AES niveau 2 : chaque ligne de programme est traduite en une séquence de mots correspondant à tous les nœuds apparaissant dans le sous-arbre associé (en rouge, bleu et noir sur la figure).

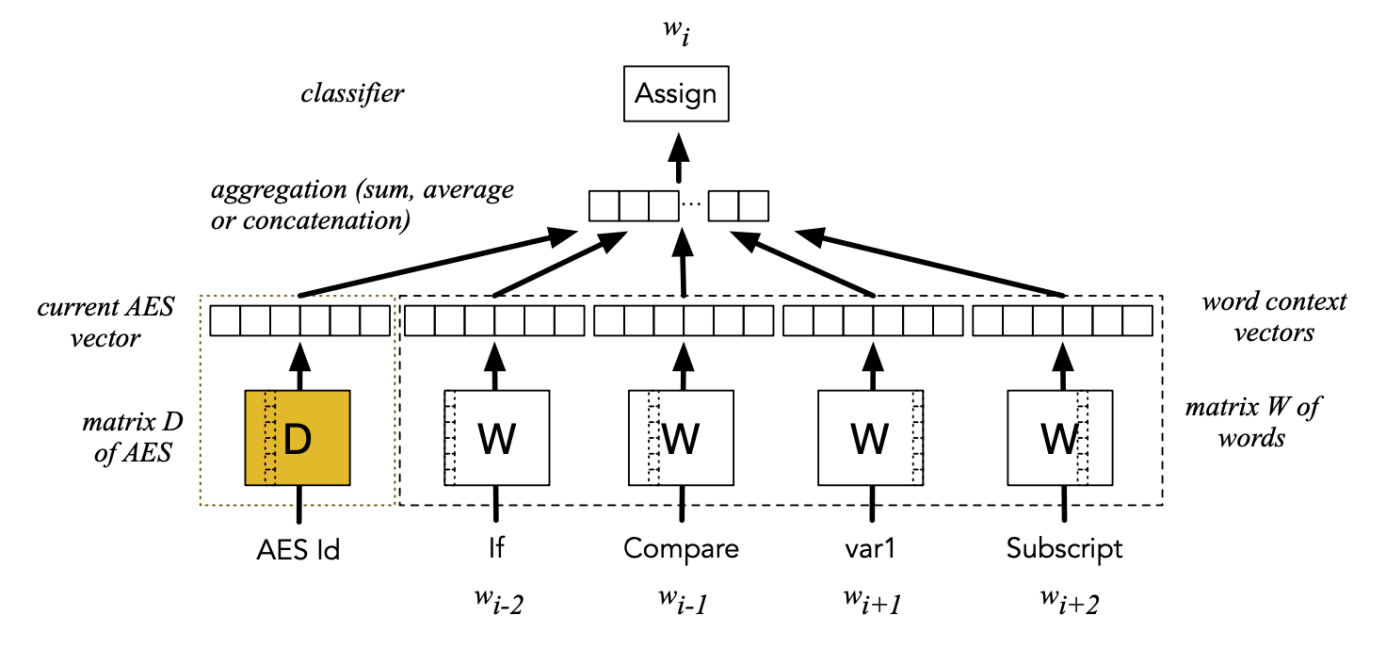
code2aes : construction de séquences d'exécution abstraites (AES) - II



aes2vec : apprentissage des intégrations de programmes d'AES - I

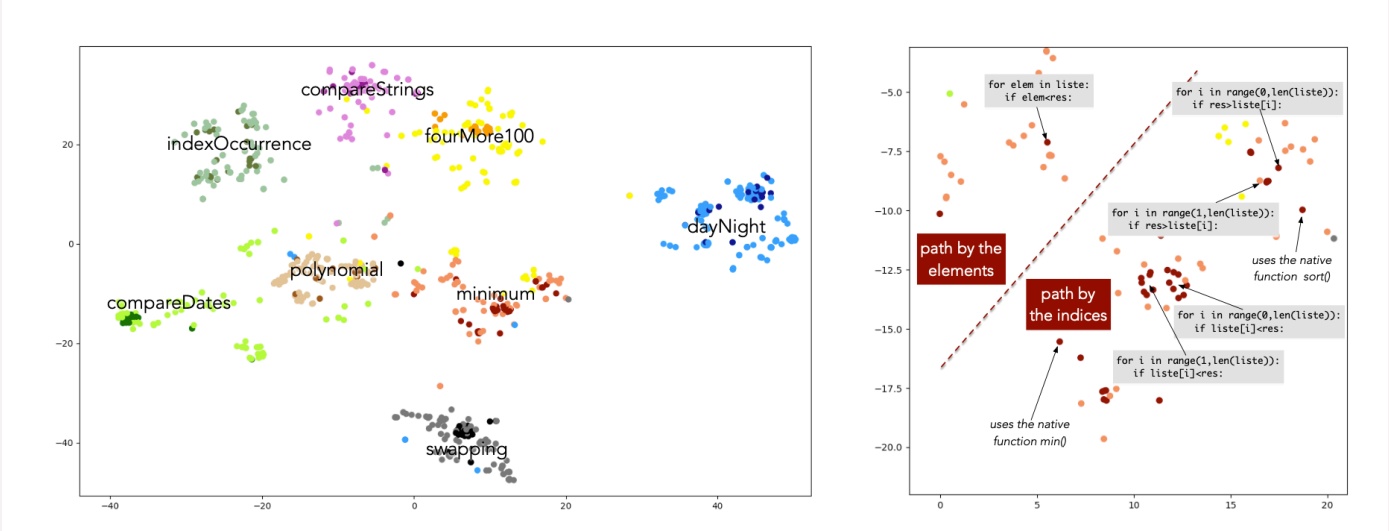
- Les programmes sont considérés comme des documents textuels dont la séquence de mots est donnée par un AES obtenu à l'étape précédente (*code2aes*).
- Dans ce modèle d'apprentissage, chaque vecteur AES est utilisé uniquement pour les prédictions des tokens de cet AES, tandis que les vecteurs de tokens (en W) sont communs à tous les AES.
- Une fois le modèle entraîné, la matrice D contient les intégrations des programmes. Les autres paramètres du modèle restant fixes (W ainsi que les paramètres softmax).
- Enfin, le choix de la stratégie d'agrégation utilisée dans la couche cachée peut être déterminant, alors qu'une stratégie de concaténation offre l'opportunité d'exploiter l'ordre des mots au sein du contexte.

aes2vec : apprentissage des intégrations de programmes d'AES - II



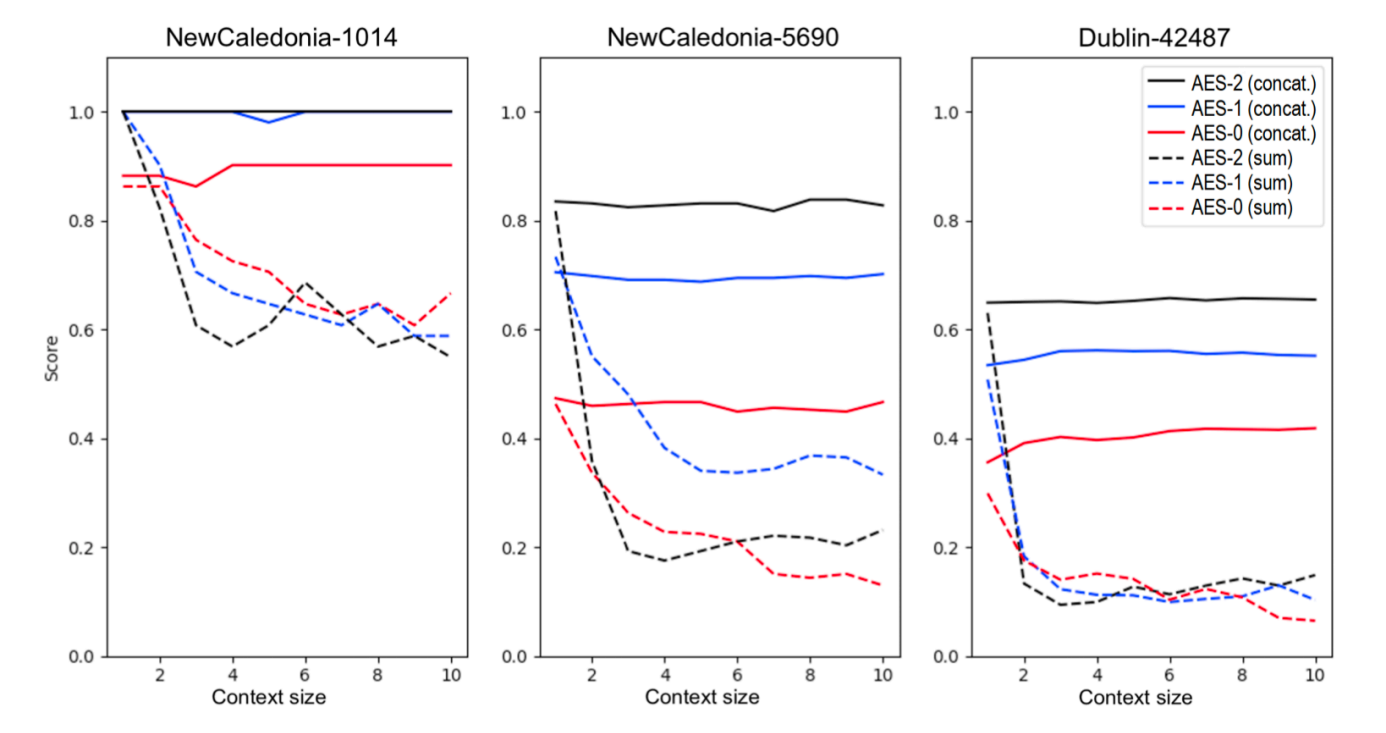
Sommaire

Évaluation de manière qualitative

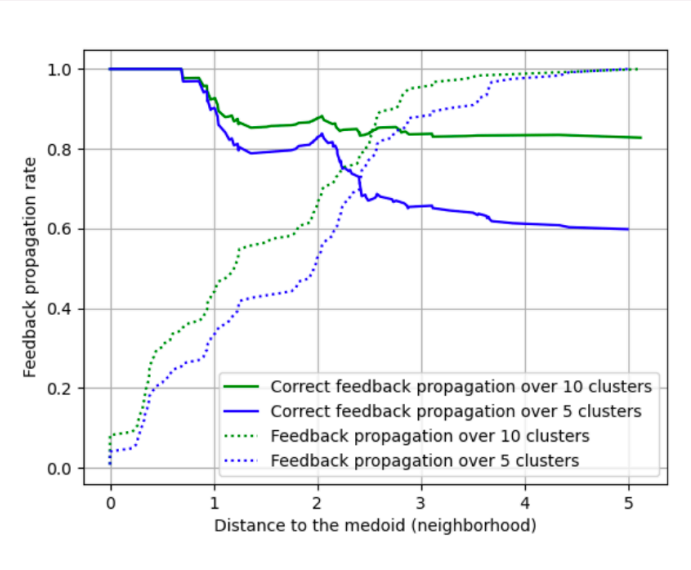


- Stylistique
- Ordre des mots
- Quelques programmes valables, mais atypiques

Évaluation de manière quantitative



Application à la propagation du feed-back



La figure montre l'évolution du taux de rétroaction correct en fonction de la taille du voisinage considéré pour la propagation. On peut voir que plus on s'éloigne des médoïdes, plus il y a d'erreurs dans les rétroactions déterminées automatiquement.

Perspectives d'avenir

- Analyser des phrases plus complexes
- Autres langages de programmation