Eötvös Loránd Tudományegyetem

Informatikai Kar

Programozáselmélet és Szoftvertechnológiai Tanszék

Kódvisszafejtés mélyhálókkal

Témavezető:
Dr. Várkonyi Teréz Anna
egyetemi adjunktus

Szerző:
Csertán András
programtervező informatikus BSc

Budapest, 2022

EÖTVÖS LORÁND TUDOMÁNYEGYETEM
INFORMATIKAI KAR

SZAKDOLGOZAT TÉMABEJELENTŐ

Hallgató adatai:
Név: Csertán András
Neptun kód: NDLG3A

Képzési adatok:
Szak: programtervező informatikus, alapképzés (BA/BSc/BProf)
Tagozat : Nappali

Belső témavezetővel rendelkezem

Témavezető neve: *Dr. Várkonyi Teréz Anna*
munkahelyének neve, tanszéke: ***ELTE-IK, Programozáselmélet és Szoftvertechnológia Tanszék***
munkahelyének címe: ***1117, Budapest, Pázmány Péter sétány 1/C.***
beosztás és iskolai végzettsége: ***adjunktus, PhD***

A szakdolgozat címe: Kódvisszafejtés mélyhálókkal

A szakdolgozat témája:
(A témavezetővel konzultálva adja meg 1/2 - 1 oldal terjedelemben szakdolgozat témájának leírását)

A kódvisszafejtő program feladata egy futtatható fájlból előállítani az annak megfelelő, magasszintű forrásfájlt. Működése a fordítóprogramok működésével ellentétes, amik egy magasszintű forrásfájlból állítják elő a futtatható állományt. Segítségével megismerhetjük a program valódi működését, ezáltál többek között kártékony szoftverek detektálására is használhatjuk.
Szakdolgozatom célja egy olyan alkalmazás készítése, ami a fenti problémát mély neurális hálók segítségével oldja meg. A program bemenete az alacsony szintű Assembly kód, kimenete pedig a magas szintű, egy átlagos programozó által is értelmezhető C nyelvű kód. A hatékonyság növelése érdekében az Assembly kód először szegmentálásra kerül, így blokkokat kapunk, amik nagyjából egy C utasításnak felelnek meg. A szegmentálást és a megfelelő C utasítás meghatározását egy-egy neurális háló fogja végezni. Ezek után a kapott C utasításokat összefűzve, valamint az esetleges hibákat korrigálva megkapjuk a várt kimenetet.
A felhasználónak lehetősége lesz saját adatokkal a háló tanítására is, valamint különböző példákon keresztül a működés kipróbálására. Az implementáció Python nyelven fog történni.

Budapest, 2021. 11. 30.

Tartalomjegyzék

1. fejezet

Bevezetés

A fordítóprogramok feladata a magas szintű programkód átalakítása a számítógép számára értelmezhető formára. Céljuk, hogy a programozó sokkal magasabb absztrakciós szinten fejezhesse ki szándékát, ezáltal megkönnyítve a szoftverfejlesztés folyamatát. A kódvisszafejtő programok működése ezzel ellentétes, az alacsony szintű, gépközeli kódot alakítják át magas szintű kóddá. Segítségével beleláthatunk a program forráskódjába akkor is, ha csak egy futtatható állomány áll rendelkezésre, ezáltal könnyebben megvédhetjük gépünket a kártékony szoftverektől.

A fenti feladat megoldására már léteznek különböző kódvisszafejtő programok [1, 2], ugyanakkor ezek legnagyobb hátránya, hogy a fordítóprogramokhoz hasonlóan bonyolult szabályok alapján működnek. Ez azért probléma, mert ezen szabályokat minden programozási nyelv esetén külön meg kell fogalmazni, ami egy bonyolult és időigényes feladat.

Dolgozatomban egy olyan programot mutatok be, ami a fenti problémát a természetes nyelvfeldolgozásban használt gépi tanulási eszközökkel oldja meg. A neurális gépi fordítás az utóbbi években hatalmas fejlődésen ment keresztül[bert], így könnyen adódik, hogy ezen eredményeket fel lehetne használni a kódvisszafejtéshez, annyi különbséggel, hogy nem angolról németre, hanem például assembly-ről C-re fordítunk.[1] Ezen módszer előnye, hogy nem szükséges hozzá programozók hosszú ideig munkája a különböző szabályrendszerek megalkotásához, valamint egy másik nyelvre való áttérés sem okoz különösebb nehézséget. Továbbá a különböző programkód generáló szoftvereknek[**??**] hála, lényegében korlátlan mennyiségű adat áll rendelkezésre.

[1] Majd látni fogjuk, hogy a programozási nyelvek sajátos szerkezete miatt sajnos nem alkalmazhatók egy az egyben a természetes nyelvek fordítása során elért eredmények.

2. fejezet

Neurális gépi fordítás

3. fejezet

Felhasználói dokumentáció

A program `Ubuntu 20.04` operációs rendszer alatt fut. A futtasához szükség van `Python 3.10.4`-re, valamint több könyvtárra, ezeket a `requirements.txt` fájl tartalmazza, melyeket a `pip` csomagkezelő segítségével könnyen feltelepíthetünk az alábbi paranccsal: `pip install -r requirements.txt`

4. fejezet

Fejlesztői dokumentáció

[3]

5. fejezet

Összegzés

Irodalomjegyzék

[1] National Security Agency. *Ghidra*. `https://ghidra-sre.org/`. 2019.

[2] Peter LaFosse Jordan Wiens Rusty Wagner. *Binary Ninja*. `https://binary.ninja/`. 2016.

[3] Xuezixiang Li, Yu Qu és Heng Yin. „Palmtree: learning an assembly language model for instruction embedding". *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2021, 3236–3251. old.

Ábrák jegyzéke

Táblázatok jegyzéke