Jegyzőkönyv

IET HF1: Szemantikus keresés

Bendicsek Márton Bendegúz

Github: https://github.com/bendicsekb/semantic\_web.git

Feladat 1 indexelés

Egyszerű indexelést pythonban dictionarykkel érdemes csinálni, hiszen az már egy indexelt adatstruktúra. A fájlokat beolvasva szavakra és fájlokra is raktam indexet.

A program használatához a create\_index.py-t a tartalmazó mappából kell futtatni (python 3.8), két kikötéssel:

az indexelendő szöveg kiterjesztése nem lehet .json

a fájlok a create\_index.py mappájától relatív “../res/indexing” mappában legyenek

A program felépít egy indexet, majd elmenti a simple\_indexes.json fájlba.

Példa az indexre (python annotációkkal):

*{*

*'words': { 'john': {'document1': 3, 'document2': 5},*

*'doe' : {'document1': 1, 'document3': 2}},  
'documents': { 'document1': {'foo':2, 'bar':1 },*

*'document2':{'asd': 1, 'bsd': 5}}*

*}*

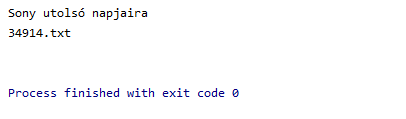
Feladat 2 kezelés

A második feladathoz írt program a már meglévő indexet beolvassa, az indításkor paraméterként kapott szavak dokumentumbeli gyakorisága alapján csökkenő sorrendben kiírja azokat a dokumentumokat melyekben minden keresőszó szerepel. Az összes szó dokumentumbeli előfordulását szummáztam dokumentumonként.

Példa:

python my\_search.py Sony utolsó napjaira

Kimenet:



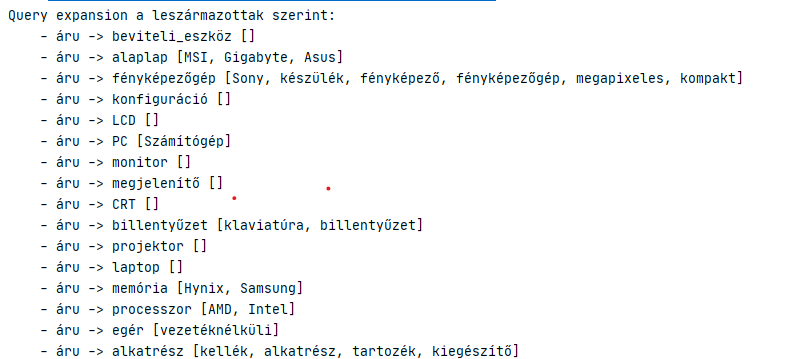
A feladat során felhasználtam a dokumentum és a szó alapú indexelést is.

Feladat 3 szemantikus keresés

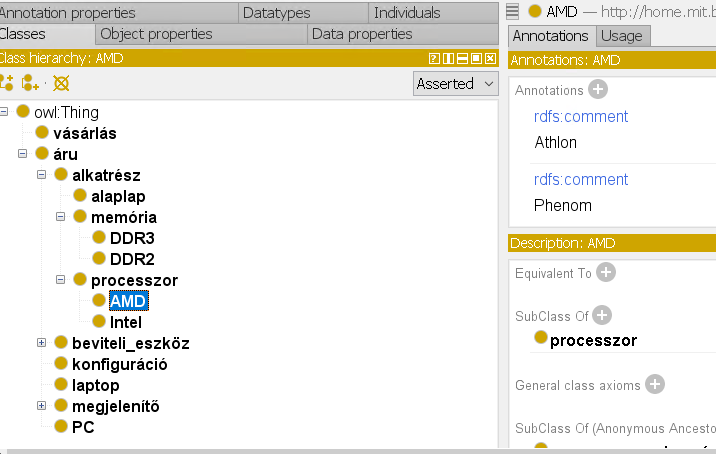
A keresőszó kiegészítés egyik formája az ontológia alapú, ebben az esetben a pc\_shop ontológiáját lehetett továbbfejleszteni, a célja hogy a keresésnél egy remélhetőleg relevánsabb (több információt tartalmazó) eredményt kapjunk.

Az ontológiában szerepelt már néhány dolog, ezt kiegészítettem egy fényképezőgép osztállyal. Ezen kívül kommentként beírtam a szinonimaszótárban található néhány megfelelőt, a fényképezőgéphez pedig a 34914.txt-ből néhány várhatóan fényképezőgépre jellemző szót.

A példakód kimenete így:



Ezután minden munkám elveszett, ezért más stratégiát választottam és a processzor és memória alkatrészeknek csináltam alosztályokat, azoknak pedig kommenteket. A példakódot úgy módosítottam, hogy az stdoutra csak a nekem szükséges Query Expansion eredményét írja ki vesszővel elválasztva.



Az ontology\_search.py ezt a lefordított java programot hívja meg és az eredményt úgy dolgozza fel, hogy ha bármelyik szó benne van egy dokumentumban akkor azt találatként adja. Ez az előző feladathoz képest jelentősen lazább követelmény, viszont így a Query Expansion lényege, hogy több találat legyen. Azért döntöttem így, mert nem ismerem a cikkeket amikben keresek, megtippelni sem tudom, hogy melyik szó lesz biztosan benne, ezért annak kommentet sem tudtam csinálni. Így egy általánosabb keresést kaptam.

Példák:

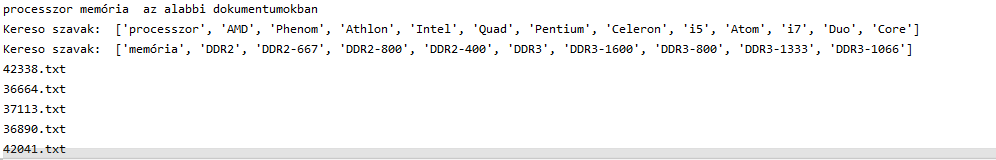
python ontology\_search.py processzor

Kimenet:



python ontology\_search.py processzor memória

Kimenet:



A lista természetesen mindkét esetben sokkal hosszabb, mint a képen.