

# Jegyzőkönyv

## IET HF1: Szemantikus keresés

Bendicsek Márton Bendegúz

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

```
Sony utolsó napjaira  
34914.txt
```

```
Process finished with exit code 0
```

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:

Query expansion a leszámazottak szerint:

```
- áru -> beviteli_eszköz []
- áru -> alaplap [MSI, Gigabyte, Asus]
- áru -> fényképezőgép [Sony, készülék, fényképező, fényképezőgép, megapixel, kompakt]
- áru -> konfiguráció []
- áru -> LCD []
- áru -> PC [Számítógép]
- áru -> monitor []
- áru -> megjelenítő []
- áru -> CRT []
- áru -> billentyűzet [klaviatúra, billentyűzet]
- áru -> projektor []
- áru -> laptop []
- áru -> memória [Hynix, Samsung]
- áru -> processzor [AMD, Intel]
- áru -> egér [vezeték nélküli]
- áru -> alkatrész [kellék, alkatrész, tartozék, kiegészítő]
```

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.

The screenshot shows an OWL ontology editor interface. On the left, a class hierarchy is displayed under 'owl:Thing'. The hierarchy includes 'vásárlás', 'áru', 'alkatrész', 'alaplap', 'memória' (with subclasses 'DDR3' and 'DDR2'), 'processzor' (with subclasses 'AMD' and 'Intel'), 'beviteli\_eszköz', 'konfiguráció', 'laptop', 'megjelenítő', and 'PC'. The 'AMD' class is highlighted. On the right, the 'Annotations: AMD' panel shows 'rdfs:comment' with the value 'Athlon' and another 'rdfs:comment' with the value 'Phenom'. Below this, the 'Description: AMD' panel shows 'Equivalent To' with a plus sign, 'SubClass Of' with a plus sign and 'processzor', and 'General class axioms' with a plus sign. The 'SubClass Of (Anonymous Ancestor)' panel is also visible at the bottom.

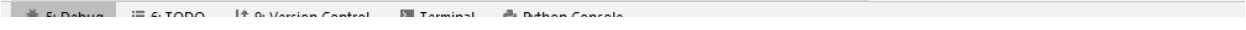
Az `ontology_search.py` ezt a lefordított java programot hívja meg és az eredményt úgy dolgozza fel, hogy ha bármelyik szó benne van egy dokumentumban akkor azt találatként adja. Ez az előző feladathoz képest jelentősen lazább követelmény, viszont így a Query Expansion lényege, hogy több találat legyen. Azért döntöttem így, mert nem ismerem a cikket 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:


```
processzor az alábbi dokumentumokban
Kereso szavak: ['processzor', 'AMD', 'Phenom', 'Athlon', 'Intel', 'Quad', 'Pentium', 'Celeron', 'i5', 'Atom', 'i7', 'Duo', 'Core']
42338.txt
36664.txt
37113.txt
36890.txt
42041.txt
36568.txt
38352.txt
37832.txt
35654.txt
37307.txt
38011.txt
42997.txt
35532.txt
35464.txt
35956.txt
42891.txt
38110.txt
38042.txt
37418.txt
42577.txt
```



```
python ontology_search.py processzor memória
```

Kimenet:

```
processzor memória az alábbi dokumentumokban
Kereso szavak: ['processzor', 'AMD', 'Phenom', 'Athlon', 'Intel', 'Quad', 'Pentium', 'Celeron', 'i5', 'Atom', 'i7', 'Duo', 'Core']
Kereso szavak: ['memória', 'DDR2', 'DDR2-667', 'DDR2-800', 'DDR2-400', 'DDR3', 'DDR3-1600', 'DDR3-800', 'DDR3-1333', 'DDR3-1066']
42338.txt
36664.txt
37113.txt
36890.txt
42041.txt
```



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