

Komplex MI rendszerek házi feladat

Problémafelvetés

A Steam nevű online játéklplatform és webshop számos keresési opciót tartalmaz, lehet például szűrni címszavakra, kategóriákra és a felhasználók által megadott címkékre. Ezeket a címkéket a felhasználók szabadon adhatják meg játékokra, és lehetséges ezekre szavazni, azonban egyéb szemantikus információ nem tartozik hozzájuk, így például a „Horror” és a „Psychological horror” címkéknek semmi közük nincs egymáshoz. Ezt a hibát kívántam egy saját ontológia felhasználásával kiküszöbölni.

Specifikáció

A projekt során egy java nyelvű programot készíték, amely letölti a Steam honlapjáról a legnépszerűbb címkéket, elkészít belőlük egy kezdetleges ontológiát, letölti Steam-ről a teljes játékgyűjteményt egy saját rdf adatbázisba, és képes ontológia alapján kiegészített keresést végezni az eltárolt játékok címkéin.

Ontológia készítés

A programot a „CreateOntology” argumentummal elindítva jutunk az ontológia készítés funkcióhoz. Futás közben a https://store.steampowered.com/tag/browse#global_492 url-t használok, ez az oldal a Steam legnépszerűbb felhasználói címkéit jeleníti meg a felhasználók számára. A letöltött oldalból kinyerem a listában szereplő címkéket, és ezek feliratai alapján készíték egyszerű OWL ontológiát, amely minden címkét egyedi osztályként tartalmaz. A megfelelő működéshez ezt az ontológiát kézzel kell kibővíteni.

Kódban átírva egy változót lehetőség van, hogy az ontológiához használt címkéket ne a weblapról, hanem egy csatolt .csv fájlból olvassa be. Ez a táblázat a <http://steamspy.com/tag/> honlapról származó 50 legtöbbet használt címkét tartalmazza, és a prototípus fejlesztésében sok időt megspórolt, hogy csak egy 50 elemű ontológiát kellett átszervezni a teszteléshez.

Ontológiaszervezés

Az elkészült ontológiát manuálisan kell átszervezni úgy, hogy az osztályok leszármazottsági viszonyai megfelelőek legyenek. Javasolt a Protege használata.

Adatbázis feltöltése

A programot a „FillDB” argumentummal elindítva jutunk az ontológia készítés funkcióhoz. A működés előfeltétele, hogy fusson egy Rdf4j szerver a localhost 8000-es portján. A teljes játék lista a (<http://api.steampowered.com/ISteamApps/GetAppList/v0001/>) Steam nemhivatalos API-járól kerül letöltésre. A lekérés eredménye egy Json objektum, ami minden Steam-en lévő artifact appld-jét tartalmazza. Ezek bizonyos része nem elérhető (például: „Dedicated server”, „Steam client”, „winui2”), így azok nem is fognak bekerülni az adatbázisba. Az innen kapott appld listán végigiterálva a játékokhoz az adatok két helyről származnak, egyrészt az előbb említett nem publikus API-ról

(<https://store.steampowered.com/api/appdetails?appids=8930>) le lehet kérni egy-egy applD-hez tartozó információkat, azonban itt nem szerepelnek a felhasználók által megadott címkék. A másik lekérés az appld-hez tartozó publikus steam oldalról történik (<https://store.steampowered.com/app/413150>), a letöltött html oldalból nyerem ki a címkéket.

Az adatbázisba {appld; ?pred; ?value} alakban kerülnek feltöltésre az adatok, ahol a ?pred a „title” „category” „tag” „type” és „metascore” értékeket veheti fel a ?value pedig a hozzá tartozó értékeket. Az adatbázisban minden IRI a <http://hu.komplexmi.e5q6ui#> névtér prefixszel van ellátva.

Keresés

A programot a „Search” argumentummal elindítva jutunk a keresés funkcióhoz, a használt ontológia jelenleg a kódban van megadva, a projekt gyökerében keres egy GamingOntology_done.owl nevű fájlt. A felület bekéri a keresett kifejezést, majd, ha az ontológia tartalmazza osztályként a címkét, akkor elkéri annak leszármazottait (ha nem tartalmazza a címkét, akkor nem egészíti ki a keresést). Az adatbázisból kikeresi azokat a appld-ket, és a hozzájuk tartozó címeiket, amik rendelkeznek az így összeállt címkével vagy címkékkel. Az eredmény a konzolon jelenik meg, három oszlopban: appld, cím, url ahol a játék elérhető.