DBPedia magyar nyelvű szövegek elemzéséhez

Németh Bottyán[1], Vándor Tamás[2]

[1]BME, TMIT,
Budapest, Magyar tudósok körútja 2., e-mail:bottyan@gmail.com
[2]Webra International Kft.,
Budapest, Francia út 33. IV/1, e-mail:tamas.vandor@webra.hu

Kivonat A publikációban egy olyan szövegannotáló rendszer kerül bemutatásra, ami a Wikipédiát is felhasználja az általa ismert fogalmak körének bővítésére. Ehhez szükséges volt a Wikipédia formális ábrázolása, amire eddig az egyik legsikeresebb kísérlet a DBPedia projekt. A DBPedia magyar változatának elkészítése után ezt a tudásbázist használtuk fel szövegek szemantikus annotálására, több más nyelvészeti eszközzel és doménspecifikus ontológiákkal kiegészítve. Így egy komplex rendszer jött létre, ami képes magyar nyelvű szövegek elemzésére és a benne található szavak szemantikus annotálására. A Wikipédiára épülő tudásbázisnak köszönhetően nagy lefedettséget, míg a formális ábrázolás miatt megfelelő pontosságot sikerült elérni. [1]

Kulcsszavak: információkinyerés, természetesnyelv-feldolgozás, ontológia, DBPedia, Wikipédia, névelem-felismerés

1. Bevezetés

Az ismertetésre kerülő magyar nyelvű szemantikus szövegannotáló rendszer elkészítése során több már korábban meglévő eszközt integráltunk egy egységes keretrendszerbe, kiegészítve saját fejlesztésű modulokkal. A munka során elért egyik legjelentősebb eredmény, hogy létrejött egy szabadon hozzáférhető formális tudásbázis a Wikipédia alapján, a DBPedia magyar változata. Ez a tudásbázis alkotja a rendszer fogalmi adatbázisának magját.

A cikkben bemutatásra kerülő rendszer egy átfogóbb projekt részét képezi. A hosszabb távú cél egy intelligens ügyfélszolgálati megoldás, amely képes egy szűkebb tárgyterületen gyakran előforduló problémákat automatikusan megválaszolni. Az ehhez vezető úton első lépésként egy olyan átfogó tudásbázis félautomatikus építését tűztük ki célul, amely kiegészítve kisebb doménspecifikus ontológiákkal az ügyfélszolgálatra érkező levelekben található fogalmakra nagyarányú lefedettséget biztosít. A következőkben az itt felhasználandó ontológia felépítéséről és első alkalmazásáról fogunk beszámolni egy magyar nyelvű szemantikus szöveganntáló rendszer keretében.

[1] A cikkben tárgyalt rendszer a "KMOP-2007-1.1.1 Intelligens Multi-Modális Tudásközpont" projekt keretében jött létre.

2. Kapcsolódó munkák

Mivel még nincsen igazán hatékony tanuló algoritmus ontológiák automatikus építésére, az ontológiák főleg kézzel készülnek (Cyc, SUMO). Egy ontológia elkészítése költséges folyamat, mint egy komoly lexikoné is. Utóbbi probléma megoldására született a Wikipédia, ami a nagyméretű közösség erejét használja fel a világ legnagyobb lexikonjának elkészítéséhez és karbantartásához. A Wikipédia az emberi tudás egy hatalmas és folyamatosan bővülő tárháza, kézenfekvő hát az ötlet, hogy a Wikipédia alapján építsünk formalizált tudásbázist, ontológiát. A nehézség, hogy a Wikipédiában található információ nagy része informális, természetes nyelvű szöveg. Ennek ellenére több ígéretes kísérlet történt az itt fellelhető tudás hasznosítására. Suchanek [6] a Wikipédia cikkeire illesztett felszíni minták alapján próbált bővíteni egy már meglévő ontológiát új tényekkel. A hamis találatok kiszűrése érdekében a talált tényeket összevetette az ontológiában már megtalálhatóakkal, és így szűrte ki az inkonzisztens találatokat. Mások [10] a Wikipédián található címszavak kategorizálásával foglalkoztak a hozzájuk tartozó cikkek tartalma alapján. Cucerzan [9] pedig a Wikipédia korpuszát használta fel névelemek egyértelműsítésére, összevetve a szövegben talált névelemek kontextusát az elemhez tartozó Wikipédia-cikk tartalmával. A DBPedia projekt a Wikipédiában található címszavakat rendezi egy ontológiába kategorizálva azokat, és az egyes cikkekből a címszavakhoz tartozó fontosabb tulajdonságokat is automatikusan meghatározzák. A tudásbázist a Wikipédián kívül más tudásbázisokban található adatokkal is bővítik a LinkedData szabvány segítségével. A DBPedia projekt ma is aktív és folyamatosan fejlődik. A nagy fogalmi lefedettség mellett a projekt köré szerveződő közösség aktivitása volt az indok, hogy a DBPedia adatbázisát választottuk az általános témájú ontológiánk alapjául.

Az ontológia építése mellett készítettünk egy többlépcsős szövegelemző rendszert, amely képes magyar nyelvű szövegekben az ontológiában tárolt fogalmak beazonosítására és így a szöveg szemantikus címkézésére. A szövegek automatikus címkézésének problémájával már többen foglalkoztak. A legtöbb megközelítés, mint a GATE [4], a PANKOW [3] vagy az ONTEA [7] mintafelismerésen alapszik. Ezen algoritmusok nehezen alkalmazhatóak kiterjedt ontológiákra, mivel a szabályrendszer létrehozása emberi erőforrást igényel. Ráadásul általában csak egyszerűbb összefüggéseket lehet jól leírni szabályok segítségével. A bonyolultabb vagy kevésbé gyakori megfogalmazások gyakran kimaradnak a szabályok közül, ezért a szabályalapú megközelítések találati aránya legtöbbször alacsony. Ezektől eltérően statisztikai módszereket és felügyelt tanulást kombináló szövegannotáló rendszer a SemTag [5]. A SemTag az ontológia elemeit a szövegkörnyezet alapján próbálja egyértelműsíteni nagy szövegkorpuszon készített statisztikák segítségével. A statisztikák készítésénél kihasználják a fogalmi hierarchiában rejlő információkat. Az algoritmus tovább pontosítható, ha kézzel címkézett tanítópéldákat is megadunk. A projekt során igen nagy mennyiségű szöveget annotáltak, viszont a felhasznált ontológia mérete kicsi volt, ezért a szöveg lefedettsége elég alacsony maradt, weboldalanként átlagosan alig több mint másfél entitást címkéztek fel.

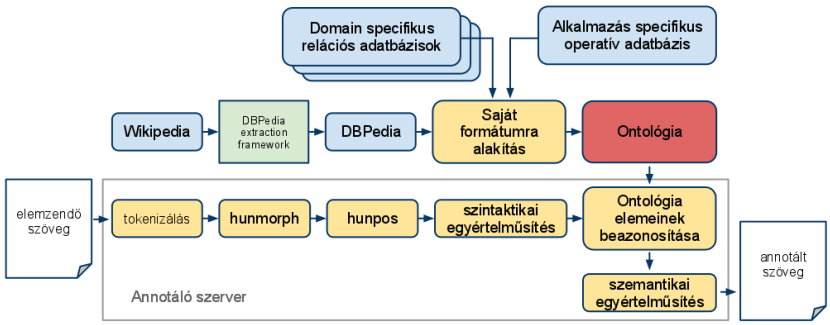
Mivel célunk volt, hogy nagyméretű ontológia alapján annotáljuk a szövegeket, a szabályalapú módszereket elvetettük, mert nagyon sok munka lenne az ontológiában előforduló összes osztályhoz szabályokat felvenni. Továbbá problémás lenne a szabályok karbantartása is, ami elengedhetetlen, ha egy olyan élő és folyamatosan változó tudásbázist alkalmazunk, ami a Wikipédián alapul. Így az egyértelműsítésnél olyan módszereket kerestünk, amik az ontológiában rejlő strukturális információk kihasználásával automatikusan oldják meg a feladatot. A projekt jelenlegi fázisában nem volt célunk a teljes egyértelműsítés, megelégedtünk azzal, hogy a lehetséges opciók számát ember számára gyorsan felfogható méretűre csökkentsük, kiválogatva néhány releváns opciót.

3. Az elemzőrendszer

Az annotálandó szövegek egy többlépcsős feldolgozási folyamaton mennek keresztül, aminek része a tokenizálás, morfológiai elemzés, POS-elemzés, névelem-felismerés és az utolsó fázisban az ontológia fogalmainak beazonosítása. A szintaktikai elemzésre szabadon felhasználható szoftvereket alkalmaztunk, mint a hunmorph [8], hunpos, valamint az OpenNLP tool maximum entrópián alapuló eszközeire [1] épülő saját elemző szoftvereket. Az elemzési lánc egyes komponensei közötti kommunikációra egy belső XML-reprezentációt alkalmazunk, ami tartalmazza az eredeti szöveget is, és az egyes modulok ezt egészítik ki egymásra épülő plusz információkkal. Az XML formátuma úgy lett kialakítva, hogy a bemenetre egyszerű szöveg vagy HTML is érkezhet és az elemzőrendszer megőrzi a kapott szöveg formázását. A külső modulokhoz készítettünk egy-egy csomagoló komponenst, ami elvégzi a transzformációt a belső XML formátum és a külső modul saját formátuma között. Az elemzőrendszer futtatható batch módban, ha egy meglévő szövegbázist kell feldolgozni és szolgáltatásként is, ahogyan használható HTML-oldalak valós idejű annotálására.

A rendszer felépítését és a felhasznált modulok sorrendjét az 1. ábra szemlélteti. A beérkezett szöveget először mondatokra, illetve szavakra bontjuk. Ezt a lépést az OpenNLP keretrendszer segítségével végeztük. Az OpenNLP alapvetően angol nyelvre készült, de csekély módosításokkal és a maximum entrópia nyelvi modellek újratanításával magyar nyelvre is alkalmazhatónak bizonyult. A szavakat ezek után szintaktikailag elemeztettük a hunmorph morfológiai elemzővel. A szövegen a hunpos POS-taggert is lefuttattuk. Mivel a hunmorph több lehetséges elemzést is előállít egy szóhoz, ki kellett választanunk az aktuális szövegkörnyezetnek legmegfelelőbb verziót. Ezt szintén egy maximum entrópia modellel oldottuk meg, ahol a kontextuális jellemzők előállításához formai jegyeket és a POS-tagger kimenetét használtuk. A jellemzők között szerepelt az aktuális szó mondaton belüli pozíciója, az aktuális, előző és következő szó végződése, illetve POS-tagje, a szó hossza és hogy kis- vagy nagybetűvel kezdődik-e. A hunmorph alternatívák leírására az elemzésben szereplő toldalékok nyelvtani jelöléseit és azok darabszámát használtuk, valamint a szótő és a teljes szó hosszának különbségét. Így elfogadható pontossággal sikerült a hunmorph által adott alternatívák közül a kontextusnak megfelelőt kiválasztani.

A szöveg annotálásakor a formai jegyek alapján jól felismerhető névelemek azonosítására, mint e-mail cím vagy telefonszám, használunk mintaillesztéses módszert is, de többnyire az ontológia alapján próbáljuk beazonosítani az egyes entitásokat. A felhasznált ontológia saját forrásokból, külső adatbázisokból is táplálkozik, és jelen esetben az ontológiába nemcsak a fogalmi osztálymodellt értjük bele és annak relációit, hanem a konkrét példányokat is. A példányok tárolása azért fontos, hogy az elemzőrendszer alkalmazásakor a világra vonatkozó tudás segítségével pontosabban meghatározhassuk a szöveg értelmét, kiegészíthessük automatikusan további információkkal. Így következtetéseket vonhatunk le, és végül a megfogalmazott kérdésekre sokkal pontosabb választ adhatunk. Ha a tudásbázisban megtalálható egy fogalom, pl. Budapest, amiről tudjuk, hogy egy város, és Magyarország fővárosa, akkor rögtön jóval több információ áll rendelkezésünkre, mintha csak a formai jegyek és a szövegkörnyezet alapján határoztuk volna meg, hogy egy városról van szó. Alapvető elképzelésünk az volt, hogy létrehozunk egy széles tárgyterületet lefedő általános ontológiát, és ezt egészítjük ki konkrét feladatokhoz kötődő speciális ontológiákkal. A különböző fogalmi rendszerek és adatbázisok összefogására létrehoztunk egy saját adatbázis-architektúrát, ami az OWL-szabvánnyal kompatibilis módon képes a fogalmak és példányok tárolására. Annak érdekében, hogy a tudásbázis alkalmas legyen valós idejű nyelvi elemzésre, ahol gyors válaszidőkre van szükség a nagyszámú lekérdezés miatt, az adatbázisréteg elé egy erre a célra kifejlesztett cache réteget is elhelyeztünk.



1. ábra. Az elemzőrendszer architektúrája.

Mivel igen sok entitást tartalmazó ontológiát használunk, fokozott problémát jelentett az annotálás során az egyértelműsítés, ugyanis egy hétköznapibb szóra akár több ezer találatot is adhat a keresés, amik közül általában egy helyes megoldás van az adott szövegkörnyezetben. Ugyan a projekt jelenlegi fázisában nem volt cél a teljes egyértelműsítés, de az mindenképpen szükséges, hogy ne szülessenek százával értelmetlen, a felhasználót felesleges adatokkal ellepő találatok. Ezt elkerülendő meghatároztuk a maximálisan megengedett többértelműségi szintet,

vagyis egy határértéket, aminél nem adhatunk vissza több találatot egy szóra. Hogy a találatokat szűrni tudjuk, relevanciaértékeket kellett rendelni hozzájuk. Ezt úgy állítjuk elő, hogy a talált entitásokra megszámoljuk a közvetlen és közvetett hivatkozásokat, és a legtöbbet hivatkozott elemek közül építjük a találati listát. Az ontológia fogalmainak keresésénél figyelembe vesszük az osztályhierarchiát és a fogalmak közötti összerendeléseket is. Tehát egy intézményre hivatkozásnak tekintjük azt is, ha a címe vagy az igazgatója neve szerepel a szövegben. Az egyértelműsítési lépéseket részletesebben a következő lista mutatja meg:

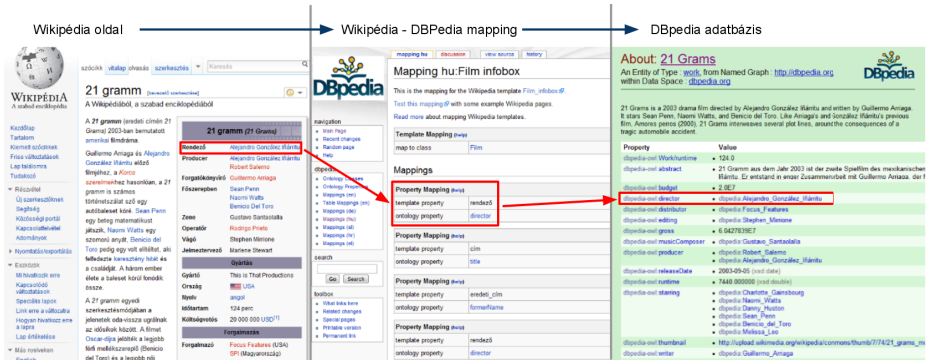
1. Az ontológiában szereplő összes entitás összes tulajdonságára megnézzük, hogy szerepel-e a szövegben. És azokat az objektumokat, amiknek valamelyik tulajdonságát megtaláltuk, felvesszük a találati listára. Ráadásul az illeszkedést nemcsak a szavak eredeti alakjára vizsgáljuk, hanem a szótövekre és többszavas kifejezésekre is.
2. Bizonyos mélységig (a kísérletekben 3-ra állítottuk ezt a limitet) felvesszük azokat az objektumokat is a találati listára, amikhez legalább az egyik talált objektum valamilyen tulajdonság mentén kapcsolódik. Pl. ha szerepel a szövegben egy utca neve, akkor az a város is, ahol az utca található, bekerül a találati listába.
3. Sorrendezzük a találati listát a referenciák alapján. Megszámoljuk, hogy melyik entitáshoz hány generáló szó volt. Egy közvetett úton felvett entitás pontszáma az őt generáló entitások pontszámának összege lesz. Ezek után meghatározzuk azt a pontszámot, amihez már kevesebb, minimum ezt a pontszámot elérő entitás tartozik, mint a maximálisan megengedett többértelműségi szint. Azokat az entitásokat, amik nem érik el a kívánt ponthatárt, eltávolítjuk a találati listából. Ha nincs ilyen pontszám, akkor elhagyjuk azokat az entitásokat, amik nem közvetlenül kapcsolódnak szavakhoz, hanem egy másik entitás generálta őket, és nincs több pontjuk, mint az őket generáló entitások közül a legtöbb ponttal rendelkező.

A fenti módszert még kiegészítettük egy kézi szűréssel, amit a doménspecifikus ontológiáknál alkalmaztunk, hogy a felszíni formákban nem megjelenő, belső, adminisztratív jellemzők, mint pl. adatbázis-azonosító, ne adjanak hamis találatokat. Itt mindössze arról van szó, hogy egyes tulajdonságoknál meg lehet adni, hogy ne keressünk rá illeszkedő szavakat a szövegben.

4. Az ontológia

Már említésre került, hogy az ontológia általános részét a Wikipédia, illetve a DBPedia alapján építettük. A DBPedia a Wikipédia legtöbb oldalán megtalálható "infobox"-ok félig formalizált tartalmát használja fel az ontológia építéséhez. Így összesen 2,6 millió egyedi entitást gyűjtöttek össze a Wikipédiáról. Ehhez társulnak a különböző doménspecifikus ontológiák, amelyek a LinkedData szabvány alapján csatlakoznak a DBPedia ontológiájához. Így az egész hálózat már 4,7 milliárd információdarabkát tartalmaz [2]. A struktúra magját képező

fogalmi hierarchia angol nyelvű, de összesen hat nyelven van összerendelés az ontológia és a Wikipédia-cikkek között, melyek egyike a magyar.



2. ábra. Wikipédia - DBPedia összerendelés

A DBPedia létrehozásához az adta az ötletet, hogy a legtöbb Wikipédia-oldalon a cikkek tartalmaznak úgynevezett "infobox"-okat. Ezek általában az oldal jobb felső oldalán megjelenő dobozok, amik táblázatszerűen foglalják össze az adott cikkben található fontosabb információkat. Az infoboxok kinézetét sablonok határozzák meg, amik témáról témára különbözőek. Így a Wikipédia-cikkeket az infobox sablonok alapján akkor is tudjuk kategorizálni, ha a Wikipédia kategóriarendszere nem megfelelő minőségű (ami sajnos igaz, különösen a magyar nyelvű Wikipédiára). Az infoboxok nem csak a kategorizálásban segítenek, hanem a sablonok tartalmazzák az adott kategóriára leginkább jellemző tulajdonságokat. A DBPedia projekt során ezeket a már-már formalizált adatokat gyűjtik össze egy RDF-en alapuló ontológiába. Az átalakítás során egyrészt automatikusan is kigyűjtik az egyes cikkekhez tartozó tulajdonságokat az infoboxokból, de van egy kézzel szerkesztett összerendelés is, ami pontosan megmondja, hogy az infoboxokban tárolt tulajdonságok minek felelnek meg az ontológiában. Erre azért volt szükség, mert az automatikus kinyerés zajos adatokat produkál. Az infoboxok nem eléggé egységes formátumúak, a Wikipédián az egyes infobox sablonok külön-külön jöttek létre, mindegyiket más-más ember szerkeszti, és így nincs igazán egységes elnevezési konvenció sem. Például semmi nem garantálja, hogy egy labdarúgónál ugyanolyan címkével jelölik a születési dátumát, mint egy festőnél. Az összerendelés karbantartására és bővítésére létrehoztak egy regisztráció után bárki által szerkeszthető szintén wiki alapon működő oldalt, ahol az infoboxok és az ontológia fogalmai közötti összerendelést meghatározó összes szabály megtalálható (http://mappings.dbpedia.org/). A projekt során mi is a kézi összerendelést bővítettük és az ez alapján létrejött ontológiát használtuk fel. A kézi összerendelésnek megvan az az előnye is, hogy a különböző nyelveken megtalálható tulajdonságok a közös fogalmi struktúrának köszönhetően egy az egyben megfeleltethetők egymásnak.

A magyar nyelvű infobox sablonok és az ontológia fogalmai közötti összerendelés elkészítésén kívül (2) kisebb módosítások az elemző program kódjában is szükségesek voltak, hogy a magyar Wikipédia-cikkek is helyesen kerüljenek bele a DBPedia ontológiájába. A magyar verzió eddig a leggyakrabban használt infobox sablonokhoz tartalmaz összerendelést. Az összerendelés a wiki elven működő oldalnak köszönhetően folyamatosan bővíthető és finomítható. Az annotáló rendszerben ezt az ontológiát kiegészítettük saját doménspecifikus adatokkal is, amelyek tartalmazzák a magyar település-adatbázist a Központi Statisztikai Hivatal településjegyzékét a települések statisztikai adataival, Magyarországon egységes kódrendszerével bővítve, valamint a Magyar Posta irányítószám-jegyzékét és a közoktatási, felsőoktatási intézmények publikusan elérhető címjegyzékét. A publikusan elérhető, hivatalos adatbázisok importálása folyamatosan történik azok kódrendszerével együtt. A kódrendszernek köszönhetően, más gépi feloldozású adatbázisokhoz is kapcsolható a rendszer.

5. Értékelés

Az eredményeket szubjektíven értékeltük több példaelemzés kézi átvizsgálásával. Jelenleg az elemzéshez felhasznált ontológia a DBPediából importált entitásokból, a magyar települések és közoktatási intézmények adatbázisából áll. Így 300 osztályt, közel 250 000 entitást és hozzájuk kapcsolódóan 510 000 tulajdonságértéket tartalmaz. A tesztek során ügyfélszolgálati leveleket és rövidebb, általánosabb témájú híreket elemeztünk. Általában a szövegekre sok illeszkedő példát talál a rendszer, és inkább a találatok szűrése okoz problémát. Sokszor magas a hamis találatok száma, de úgy gondoljuk, hogy az egyértelműsítő algoritmus fejlesztésével ezen sokat tudunk javítani. A rendszer teljesítményét a következő egyszerű példamondattal szemléltetnénk: "A Szilágyi Erzsébet Gimnázium tanulói gyakran hallgatnak Rockzenét, például GunsNRosest." A találatokat pedig a következő táblázat mutatja:

Illeszkedő szó	Tulajdonságok	Talált példányok
A	Settlement.VehicleCode	Augsburg
A	Athlete.Nationality.VehicleCode	Gregor Baumgartner, Julia Schruff
A	Person.BirthPlace.VehicleCode	Alexander Grimm
A	Country.VehicleCode	Austria
Szilágyi	last_name	SZILÁGYI, SZIL
Szilágyi	first_name	SZILÁGYI
Szilágyi	Settlement.name	Szil
Szilágyi	address.settlement.name	Dózsa György u. 1. Szil
Erzsébet	Settlement.name	ERZSÉBET
Erzsébet	first_name	ERZSÉBET

Illeszkedő szó	Tulajdonságok	Talált példányok
Erzsébet	address.settlement.name	Erzsébet Általános Iskola
Gimnázium	School.CampusType	Móra Ferenc Gimnázium Szerb Antal Gimnázium Eötvös József Gimnázium
hallgatnak	last_name	HALLGAT
Rockzenét	MusicGenre.Foaf:name	Rockmusic
GunsNRosest	Band.Foaf:name	Guns N Roses
Szilágyi Erzsébet	address.postal_name	1016 Mészáros u. 5-7. Budapest Szilágyi Erzsébet Gimnázium
Szilágyi Erzsébet	edu_institute.name edu_institute.short_name	Budapest Szilágyi Erzsébet Gimnázium

6. Összegzés

A munka során több különböző eszközt olvasztottunk egy egységes rendszerbe, kiegészítve saját fejlesztésű komponensekkel. Emellett egy magyar nyelvű ontológia építésével is foglalkoztunk, és ennek keretében elkészítettük a DBPedia magyar változatát. Az eddigiekben vázolt rendszer mélységében és funkcionálisan is fejlesztés alatt áll. A meglévő részek pontosságának fejlesztésénél a legfőbb pont az egyértelműsítés javítása a szintaktikai elemzés és főleg az ontológia elemeinek felismerése terén. Az egyértelműsítésnek kiterjedt irodalma van, és sok lehetséges megoldást vázoltak már különböző kutatók. Ezek közül olyan módszereket szándékozunk alkalmazni, melyek nem igényelnek emberi beavatkozást, és így kezelni tudjuk a nagyméretű tudásbázist. Tervezzük a meglévő ontológia finomítását és a rendszer kiegészítését úgy, hogy ne csak statikus entitásokat, hanem folyamatokat és azok állapotát is képes legyen kezelni. Fontos az is, hogy relációs operatív adatbázisokkal is összekapcsolható legyen a rendszer. Ebben az esetben már nem egy konstans, hanem egy állandóan változó tudásbázissal van dolgunk, aminek hatékony kezelése további kihívásokat rejt magában.

Hivatkozások

1. Adwait Ratnaparkhi: Maximum entropy models for natural language ambiguity resolution PhD thesis, University of Pennsylvania, 1998
2. Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, Sebastian Hellmann: DBpedia – A Crystallization Point for the Web of Data. in Journal of Web Semantics: Science, Services and Agents on the World Wide Web, Issue 7, Pages 154–165, 2009.
3. Cimiano P., Ladwig G., Staab S.: Gimme' the Context: Context-Driven Automatic Semantic Annotation With C-Pankow. in proc. of the 14th International Conference onWorldWideWeb, New York, NY, USA. ACM Press, 2005, ISBN 1-59593-046-9, pp. 332–341.

4. Cunningham H., Maynard D., Bontcheva K., Tablan V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. in proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL '02), Philadelphia, 2002.
5. Dill S., Eiron N. et al.: A Case for Automated Large-Scale Semantic Annotation. Journal of Web Semantics, 2003.
6. Fabian M. Suchanek, Mauro Sozio, Gerhard Weikum: SOFIE: A Self-Organizing Framework for Information Extraction 18th International World Wide Web conference (WWW 2009), Madrid
7. Laclavík Michal, Seleng Martin, Ciglan Marek, Hluchy Ladislav: ONTEA: Platform for pattern based automated semantic annotation Computing and Informatics, Vol. 28, 2009, 555–579, V 2009-Sep-16
8. Trón Viktor, László Németh, Péter Halácsy, András Kornai, György Gyepesi, and Dániel Varga: Hunmorph: open source word analysis in proc. of ACL., 2005
9. Silviu Cucerzan: Large-Scale Named Entity Disambiguation Based on Wikipedia Data in proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 708–716, Prague, June 2007.
10. Zeno Gantner,Lars Schmidt-Thieme: Automatic Content-based Categorization of Wikipedia Articles in proc. of the 2009 Workshop on the People's Web Meets NLP, ACL-IJCNLP 2009, pages 32–37, Suntec, Singapore, 7 August 2009.