

MONDAT FÜGGŐSÉGI GRÁFOK ONTOLÓGIAI MODELLEZÉSE

Rostás László

hallgató, Miskolci Egyetem

Informatikai Intézet, Általános Informatikai Tanszék

3515 Miskolc, Miskolc-Egyetemváros, email: kovacs@iit.uni-miskolc.hu

Kovács László

egyetemi tanár, Miskolci Egyetem

Informatikai Intézet, Általános Informatikai Tanszék

3515 Miskolc, Miskolc-Egyetemváros, email: kovacs@iit.uni-miskolc.hu

Absztrakt

A szabadszöveges dokumentumok feldolgozásának egyik legnagyobb kihívása a szöveg jelentésének feltárása. A szemantikai tartalom kibányászásához egy többlépcsős feldolgozási lánc végrehajtása szükséges. Első lépésekben a szöveg előfeldolgozási, nyelvtani, morfológia elemzések hajtódnak végre, s a folyamat végeredménye egy szemantikai, ontológia leírás. A dolgozat elemzi a függőségi gráfok szerepét ebben a folyamatban és azt vizsgálja, hogy a függőségi gráfok milyen módon konvertálhatóak szemantikai leírásra. Az elkészített mintarendszerben a gráfok RDF triplet formátumú leírásra konvertálódnak.

Kulcsszavak: *NLP, függőségi gráf, szemantikai modellek, ontológia*

Abstract

On the field of natural language processing, one of the hardest task is to implement an efficient text to semantic engine. The semantic content is generated from the free text document using a chain of processing phases. The first processing steps relate to text preprocessing, morphology and grammar analysis, while the result is a semantic graph usually stored in ontology format. The paper presents an analysis about the role of dependency graph in this conversion process. The work contains proposals for a semantic model suitable for straightforward conversion from the dependency graph. The implemented prototype system uses RDF-based ontology model to store the final semantic model.

Keywords: *NLP, dependency graph, semantic model, ontology*

1. Függőségi gráfok

A szabadszöveges dokumentumok tartalmi elemzése során a bemenő szöveget fogalmakra és a fogalmak közötti kapcsolatokra kell leképezni. Ezen témakörrel a számítógépes nyelvészet [1], a szövegbányászat [2] és a szemantika feltárás (Text to Semantic) [3] területei foglalkoznak. A szemantikai leírás segítségével további tudási-intenzív alkalmazások dolgozhatók ki, mint például a beszéd alapú ember-gép interfész vagy a szöveg alapú információ begyűjtés.

A probléma bonyolultsága több részproblémakör létrejöttét hozta magával, beleértve többek között a szöveg szegmentálása, a tulajdonnevek kiemelése, a véleményelemzés és a szöveg kivonatolása témaköreit. A szöveges dokumentum szemantikai leírásra történő konvertálása során több rétegen keresztül halad a megmunkálás. A rendszerek a következő modulokat foglalják magukba [4]:

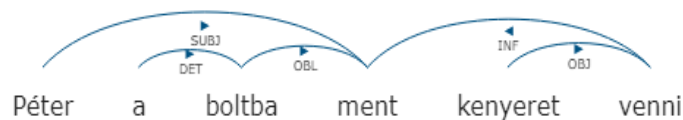
- szöveg előfeldolgozás (szegmentálás, konverzió),

- mondatok meghatározása,
- szavak meghatározása,
- szavak morfológiai elemzése,
- szavak közötti kapcsolatok meghatározása,
- mondatok közötti kapcsolatok meghatározása,
- szemantikai modell előállítás.

A feldolgozás egyik fontos lépése a szavak közötti kapcsolatviszonyok feltárása. A kapcsolatok megadásának klasszikus formáját a szavak közötti függőségi gráfok jelentik [5]. A Dependency Grammar területének közvetlen gyökereit Tesnière [6] munkájához szokás kötni. A modell alapfeltevése, hogy a mondat szavai között bináris, aszimmetrikus kapcsolatok élnek. A kapcsolat egy szintaktika és rendszerint szemantikai függőséget hordoznak. Minden kapcsolatban van egy független (head) rész és egy függő, kapcsolódó rész. Például a

Péter kenyeret eszik

mondatban az eszik, mint head részhez kapcsolódik a Péter és a kenyeret rész. Az 1.-es ábra egy összetettebb kapcsolatot ábrázol. A modellben az egyes élekhez szemantikai jelentést, tulajdonságokat szoktak rendelni. Ilyen élcímke utal többek között a tárgy vagy alany jelentésű kapcsolatokra.



1. ábra. Minta függőségi gráf

A gráf alapú mondat ábrázolásnál a függőségi kapcsolatokat az alábbi alkategóriákba osztályozhatjuk, ahol H jelöli a head részt, D a függő elem és C a nyelvtani szerkezet:

- H meghatározza C-t és önmagában is elegendő C-hez,
- H meghatározza C-t és D a C pontosítását adja,
- H kötelező, D opcionális,
- H határozza meg, hogy D kötelező vagy opcionális-e.

A függőségi gráf mechanizmus intenzíven vizsgált és alkalmazott módszer a számítógépes nyelvészet területén. Elterjedtsége ellenére van néhány olyan jellemzője, mely többértelműséget vagy funkcionális hiányt eredményez. Ilyen például az összetett logikai szerkezet esete (coordination problem [7]). Egy "A vagy B" kifejezés esetében ugyanis nem egyértelmű, hogy mely elem legyen a head rész. Lehetne ugyanis az 'A' szó és a 'vagy' szó is.

A meglévő kisebb hiányosságai, pontatlanságai ellenére, ez az ábrázolási forma jelenti a legjobb kiindulási alapot egy szemantikai kapcsolat feltáró módszer esetében, hiszen a függőség sok esetben egyértelmű szemantikai kapcsolatra is utal.

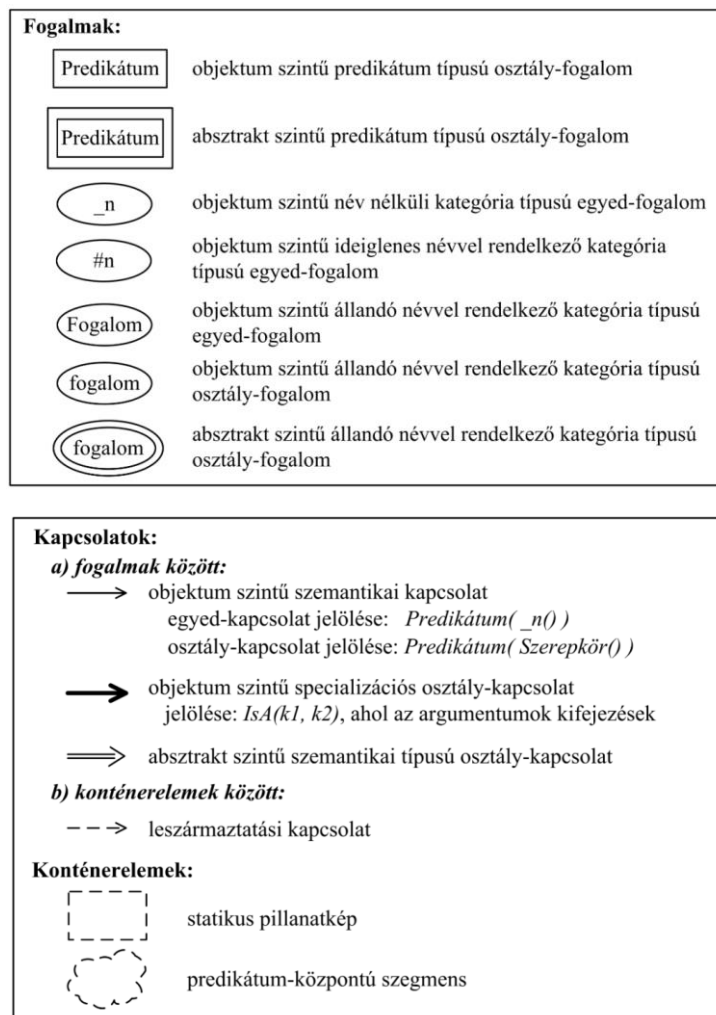
2. Szemantikai leíró modellek

A szemantikai tartalmat leíró gráfok első közvetlen elemzése Sowa [15] modelljéhez kötődik. A Sowa's Conceptual Graph (CG) modellje ötvözi a gráf és logikai formalizmusokat és alkalmas a fo-

galmak, osztályok, kapcsolatok és kvantorok ábrázolásra is. A nyelvészet oldaláról nézve a gráf élei az ígét köti össze annak kapcsolódó környezetével [16].

Ezen modellből kiindulva később számos kiterjesztett modell jött létre [17]. A dolgozatok az időbeliség, a számosság és a csoportok kezelésére javasolnak további modellelemeket. Később [18]-ban, a CG formalizmust default taxonómiai tudáselemek kezelésére is alkalmassá tették. A CG modellen alapszik a programozás-fogalmi gráf is, melyben a procedurális elemek is helyet kaptak az aktor szereplők bevonása mellett [19]. Ezen modell már alkalmas az üzleti folyamatok vizsgálatára is [20]. A fejlesztések egyik sajátos iránya a bizonytalansági elemek bevonására irányult, ennek keretében jöttek létre a fuzzy CG modellek is [21].

A kidolgozott szemantikai modellünk az ECG [Extended Conceptual Graph] modell [22] továbbfejlesztése. Az ECG modell a predikátum logika formátumában adott állítások reprezentálását tűzte ki célul. Az ECG modell építőkövei (lásd 2. ábra) a fogalmak, a fogalmi kapcsolatok és a konténerok.



2. ábra. ECG modell építőkövei

A konténer elemek szerepe, hogy általuk egy összetett hierarchia rendszer is kiépíthető legyen. A hierarchia szerepe, hogy a valóságban is megjelenő strukturáltság leképezhető legyen a számítógépes környezetre is. Az ECG modell logikai vetülettel is rendelkezik, elemeiben a predikátum logikai eszközeire épít. A modellben a predikátumok, igék játsszák a főszerepet. Az egyes ECG egységek közép-pontjában ezen predikátumok állnak. A predikátumhoz kötődnek a további leíró fogalmak, melyeknél a modell figyelembe veszi annak időbeliségét, absztrakciós szintjét, számossági jellegét is. A modell grafikus elemeit a 2. ábra foglalja össze.

A kidolgozott jelentés leíró szemantikai gráf a következő tulajdonságokkal rendelkezik:

- csomópontok fogalmak, melyeket egy vagy több szó együttese írhat le,
- egy csomópontoz tartozó szavall a mondat különböző részeiből is származhatnak,
- központi eleme a predikátumot (állítmányt) leíró csomópont,
- az élek attribútumai a mondatrészek szerepéhez kötődnek,
- vannak kötelező (pl. alany) és opcionális (pl. időhatározó) csomópontok,
- egy csomópont mögött egy al-gráf is elhelyezkedhet (pl. almondat ábrázolása).

3. Ontológia modellek

A szemantikai gráfok területéhez szorosán kapcsolódik az ontológia területe. A mai értelemben vett számítógépes ontológia fogalmának meghatározása alapvetően Gruber [8] nevéhez kapcsolódik, mely szerint az ontológia a számítógéppel támogatott egységes fogalombázis, tudásbázis kezelő rendszerek explicit eszközenszere.

Funkcionális oldalról az ontológiai rendszerek kiterjednek a fogalom alapú modellek kezelésére, a magasabb szintű fogalmak generálására, logikai következtetések végzésére, rendszer integritás ellenőrzésére és az új tények feltárása. Az ontológia több szinten is megjelenhet, több különböző ontológia szegmenst szokás megkülönböztetni. A felső ontológia (upper ontology) [9] az absztraktabb, általános fogalmak rendszerét fogja össze. A szakterület specifikus ontológia (domain ontology) ezzel szemben egy szűkebb terület tudástérképét írja le. Szokás néha a hétköznapi ontológia (commonsense ontology) területével is külön foglalkozni, mely a hétköznapi tapasztalatokból eredő tudást fedi le. Az ontológia tipikus alkalmazási területeit jól összefoglalja össze Niles cikke [10], mely alapján a következő gyakorlati célokat emeljük ki:

- az információ ábrázolás egységes formája, mely elosztott hozzáférést is támogat,
- lehetővé teszi az információk újrahasonosítását,
- irányíthatja a döntési folyamatokat,
- ellenőrizheti a rendszerek integritási egységét.

Az ontológiai adatbázisok egyik fontos eleme a szabvány adatmodell. Az adattárolás egyik elterjedt eszköze az RDF [11] nyelv, melyben az elemi állítások, tripletek, az alapvető építő kövek. Az egyes elemi állításokban fogalmak és konstansok szerepelhetnek, ahol a fogalom lehet egyed szintű és kapcsolat szintű. Az RDF adatbázis egy hálót alkot, melyben minden fogalom egy csomópont vagy él, a fogalmak globális értelmezésűek. Az RDF adatbázis az alábbi építőkövekből állítható össze:

- resource: azonosítóval ellátott egyedli fogalom,
- literal: konstans érték,
- statement: állítás,
- properties: fogalmak jellemzői (kapcsolatokkal ábrázolható).

Az RDF az ismeretek elemi mondatokon keresztüli ábrázolásán alapszik. Az összetett állításokat felbontja elemi, csak három tartalmi elemet tartalmazó állításokra. A triplet az alábbi komponenseket foglalja magába:

- `rdf:subject` – az RDF állítás alanya,
- `rdf:predicate` – az alany egy tulajdonsága,
- `rdf:object` – az RDF állítás tárgya, a tulajdonság értéke.

Mint látható, az RDF is egy elemi függőségi gráfot épít fel a jelentéstartalom ábrázolására. Az RDF korlátai közé tartozik, hogy ez a modell viszonylag szerényebb lehetőségekkel rendelkezik az

- integritási elemek,
- kapcsolat típusok,
- összetett struktúrák,
- információfeldolgozás

területein. A hiányosságok kiküszöbölésére jött létre az OWL [12] szabvány, melyben számos OOP alapú funkció beépítésre került. Emellett az OWL gazdagabb szabályrendszer leírást is támogat. Az OWL rendszer szorosan támaszkodik a logikai rendszerekre is. Az OWL rendszerben az ontológia leírása az alábbi atomokon nyugszik:

- osztály fogalom,
- skalár értékű tulajdonság,
- fogalom értékű tulajdonság,
- példány,
- szabályok.

Az OWL nyelv a (leíró logika) DL [13] nyelvvel áll szoros kapcsolatban. A DL nyelv a predikátum logikai formalizmus és a halmazelmélet integrációjának tekinthető, melyben minden fogalom mögött a hozzá tartozó példányok halmazát kell látnunk. A DL egy véges kifejezésekkel dolgozó formalizmus, melyben hatékonyabban végrehajthatóak az alapvető logikai és következtetési műveletek is.

A parancsszintű interfész mellett a legtöbb ontológia adatbázis biztosít API felületet is. Az OWL API segítségével a kliens programokból is meghívhatóak az ontológia kezelő parancsok. Az API interfész alapvetően elemi, navigációs műveleteket tartalmaz, de egyes kiegészítései támogatják a SPARQL vagy DL alapú lekérdezéseket is.

Az ontológia adatbázisok implementációjának három fő vonulata van:

- RDF triplet adatbázisok,
- OWL-XML leíró állományok,
- más adatbázis tárolók (rendszerint relációs vagy hálós).

Az RDF triplet alapú tárolás előnye a kiemelkedő hatékonyság a nagytömegű adatforrások esetében [14]. Az adatkezelések gyorsaságának növelését komplex indexelési mechanizmus támogatja. Az ingyenes RDF triplet-store adatbázisok közül az Apache Jena rendszer széles skálázhatósággal rendelkezik.

4. Függőségi gráf, mint szemantikai-ontológiai modell

Az ontológiai háttér adatbázisban történő tárolásához célszerű a leíró függőségi-szemantikai gráfokat RDF tripletek alakjában tárolni. Közvetlenül nem használható a függőségi gráf, mint adatmodell, mivel a gráf csak a mondatban található szavaknak a nyelvtani szerepét (alany, állítmány, határozók, stb), illetve a szavak egymáshoz való viszonyát adja meg, de például a rejtett alanyt és állítmányt nem is

tudja meghatározni. A tripletek előállítására emiatt egy meghatározott logikára épül, ami a lehetséges nyelvtani elemek kombinációjához rendel összerendelési szabályokat. Három fő lépésből áll a feldolgozás:

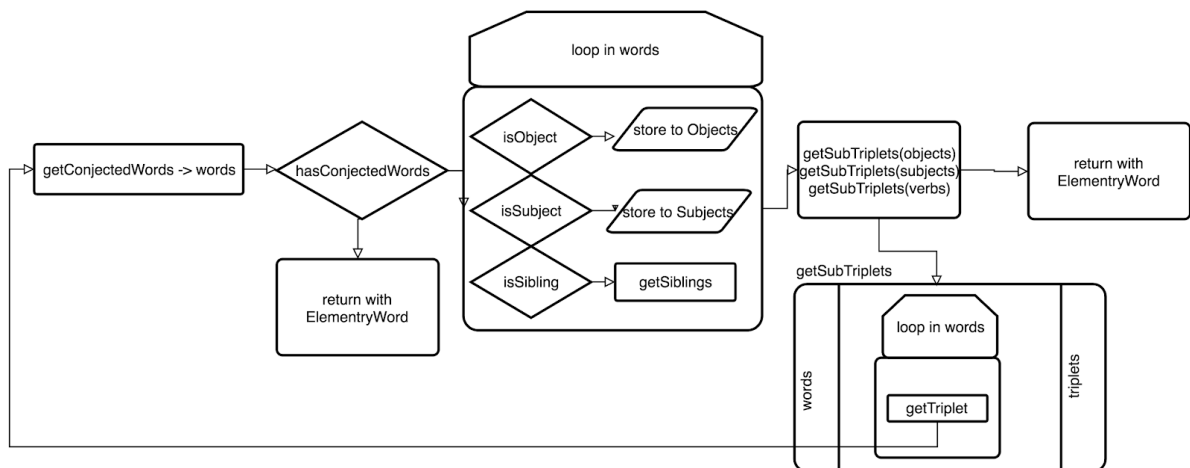
- egy saját, jelentést leíró szemantikai gráf elkészítése,
- szemantikai gráfból tripletek készítése,
- tripletek a szükséges formátumba történő konvertálása (pl.: RDF).

A módszer bemutatásához vegyük a következő egyszerű mintamondatot: „Lány tejet és vizet iszik”. A leírásban kitérünk a komplexebb mondatokra is.

id	feature	form	lemma	phheads	pos	targetId
1	Case=Nom Number=Sing	Lány	lány	SUBJ	NOUN	5
2	Case=Acc Number=Sing	tejet	tej	OBJ	NOUN	5
3	_	és	és	CONJ	CONJ	2
4	Case=Acc Number=Sing	vizet	víz	COORD	NOUN	3
5	Definite=Ind Mood=Ind Number=Sing Person=3 Tense=Pres VerbForm=Fin Voice=Act	iszik	iszik	ROOT	VERB	0
6	_	.	.	PUNCT	PUNCT	0

3. ábra Mintamondat nyelvtani elemzése

A modell felépítése során a függőségi elemzésből kiindulva alakítható ki a modell. Első lépés a ROOT elem megkeresése (példában: iszik szó), majd ehhez fog kapcsolódó gyerekelemeket generálni. A rendszer egy rekurzív fabejárás segítségével építi fel a modellt. A 4. ábra ezt a belső rekurzív folyamatot szemlélteti. A getConjectedWords modul meghatározza, hogy egy adott elemhez vannak-e hozzá kapcsolódó további szavak a függőségi elemzésben.



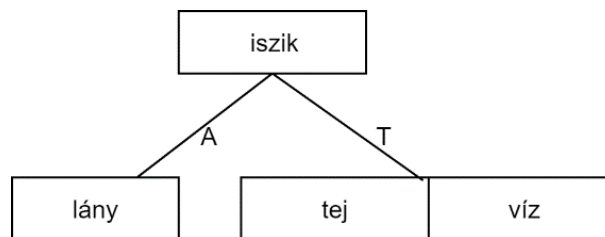
4. ábra. UML folyamatábra a getTriplet metódushoz

Példát tekintve a ROOT elemhez a lány és a tej szavak kapcsolódnak. Ha a vizsgált mondatban vannak kapcsolódó elemek, akkor meghatározza a tárolási irányt, ami lehet tárgy, alany, állítmány vagy testvérelem. Ennek meghatározásához ad segítséget a nyelvtani elemzésben a feature Majd a kapcsolódó szavakhoz is meghívja a getconectedWords elemzést. Ezzel létrehozza a rekurziót, ami addig tart, amíg talál kapcsolódó szavakat. A mintamondatban az inni lesz a mondat állítmánya, amelyhez kapcsolódik a lány szó, mint alany.

A jelen példában találhatóak testvérelemek is (tej és a víz). Ebben az esetben található egy „és” kötőszó, amely a nyelvtani elemzést tekintve mivel minden szót egy-egy token-nek tekintve nyelvtani függőségbe állít, ezért az elemzésben csak közvetve fedezhető fel. Ebben az esetben a függőségi bejárás út a tej → és → víz lesz. A mintában nem szereplő, de előforduló vesszővel való felsorolás esetében a nyelvtani függőségi fa elemzését megvizsgálva látható, hogy a vesszővel való elválasztás esetében a függőségi elemzés tényleges szülő gyerek kapcsolata megfelelő. A testvérelem meghatározása is rekurzív módszerrel történik, mivel így lehet a többelemű felsorolást feltárni.

Vannak a mondatban egyéb szereppel rendelkező szavak is. Ilyen például a melléknév, ami az állítmánynak, tárgynak vagy alanyhoz kapcsolódik. Ebben az esetben ezeket különálló tripletekként kell kezelni. Lány szépen iszik. Ekkor már az állítmány egy összetett elem lesz, ami újabb szemantikai tripletet tartalmaz. Ebben az esetben egy generált állítmány van, a lenni, mivel valaminek a milyenségét határozza meg, így az „inni lenni szép” hármast kapjuk.

Az első lépésben létrehozott szemantikai modellből (5. ábra) konverzióval létrehozhatóak a letárolandó elemi tripletek. Az RDF tripletek előállításához a mondatleíró hierarchiát fel kell bontani elemi hierarchiákra, meghatározva az elemi állítmányokat és a kapcsolódó elemeit.



5. ábra. Függőségi fa modularizálása

Az elemi egyedek és állítmányok feltárása is egy rekurzív bejárást igényel a szemantikai modellen. Jelen esetben a lány: subject, tej: object, víz: object, iszik: subject lesz az a lista, amelyek a tripletekben felhasználható. A mondat leírásához szükséges tripletek előállításánál minden konténer elemhez kell keresni alanyi és tárgyi elemet, illetve összetett, halmozott állítmány esetén az állítmány előállítására is szükség van. Ha egy elem gyereke összetett konténer elem, akkor tovább kell rekurzívan keresni, amíg elemi elemet nem találunk. Mellérendelő esetben a mondat triplet leírójának generálása során minden lehetséges variációt elő kell állítani. Pl.: Éva csinos és szép lány. – Itt a csinos és szép a mellérendelő kapcsolat és ebben az esetben két variáció van: Éva lenni csinos; Éva lenni szép. A mintamondatban a tárgyi ágon volt egy felsorolás, így ott kell több variációt létrehozni. Így a következő két RDF triplet fog generálódni:

```

:lány rdf:type owl:NamedIndividual ,
:subject .
:víz rdf:type owl:NamedIndividual ,

```

```

:object .
:tej rdf:type owl:NamedIndividual ,
:object .

:iszik rdf:type owl:ObjectProperty ;
rdfs:domain :subject .

:lány :iszik :víz .
:lány :iszik :tej .

```

Az elkészült RDF leírásban az első sorok definiálják az elemzett mondatban szereplő fogalmakat, egyedi azonosítókat rendelve hozzájuk. Az állítmány az RDF nyelvben tulajdonságként (property) jelenik meg. A leírás végén a feltárt elemi fogalmi kapcsolatok (triplet) kerültek megadásra. Az elkészített leírás szabvány eszközzel beépíthető az RDF ontológiai adatbázisokba.

5. Mérési eredmények

A kidolgozott rendszer egy külső nyelvtani elemző (e-magyar.hu) által generált függőségi gráf RDF leírásra történő konvertálását végzi el. A konverziós modul elemzésénél az egyik fontos szempont a működés helyessége és kifejező ereje. A rendszer kimenetének minősége nagyban függ a bementként kapott függőségi gráf helyességétől. Az elvégzett kísérletek alapján a függőségi gráf egy tagmondatból álló esetekben végig helyes eredményt adott, a tripletek előállítás is egyértelmű és megfelelő eredményt adott. Összetett mondatok esetében, a mondatok közötti kapcsolatok feltárása már nem minden esetben volt megfelelő. Ennek okai között szerepel a bejövő függőségi gráf ábrázolási és előállítási korlátai is.

A rendszer időbeli hatékonyságának elemzésére a mondatok feldolgozási idejét vizsgáltuk a mondatok komplexitásának függvényében. A mérésekben a rendszer inicializálása nem került mérésre, itt kizárólagosan a konvertálási időt vesszük figyelembe. Mivel a modulban a rekurzív hívások száma a fő időfaktor, ezért a vizsgált mintában négy különböző komplexitás szintet hasonlítottunk össze. A mondatok a hosszukban és a kapcsolati komplexításban is eltérnek egymástól:

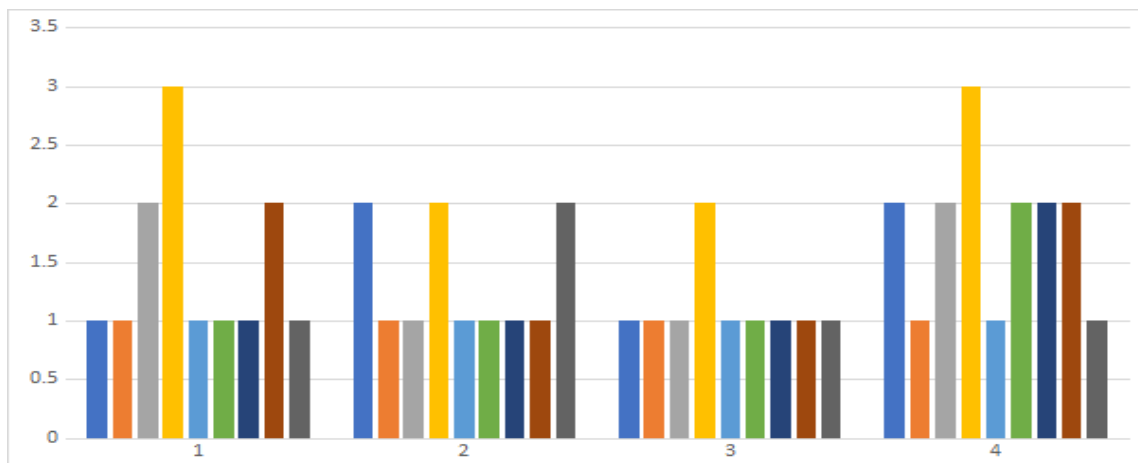
1. szint: Az aranyos lány nagy fekete szemüveget használ.
2. szint: Az aranyos lány nagy fekete szemüveget használ és iszik.
3. szint: Az aranyos lány nagy fekete szemüveget használ és az ügyes fiú tejet iszik.
4. szint: Aranyos Éva, kedves Bea egy hosszú könyvet és népszerű újságokat olvasnak.

A méréseket többször futtatva, a kapott eredményeket az 6. ábra foglalja össze. Mint látható, az elemzési idő is 1 és 3 millisec közé esik. A negyedik mondat esetében viszont egyértelműen az is látszik, hogy a több mellérendelt szó miatt a rekurzív hívások száma is megnő, így az átlagos feldolgozás is lassabb egy millisec-el.

5. Összefoglalás

A dolgozat bemutatja a függőségi gráfok szerepét a szabad szöveges dokumentumok szemantikai feltárásának a folyamatában. Kidolgozásra került egy olyan algoritmus, amely alkalmas a függőségi gráf ontológia alakra történő konvertálására. Az elkészített mintarendszer a gráfokat RDF triplet formátumú ontológia alakra fordítja át. A kidolgozott rendszer az elemi mondatok esetében a kísérletek alapján

helyes RDF ontológia leírást generál. Az elkészült ontológia szabványos eszközökkel beépíthető más meglévő ontológia adatbázisokba. A vizsgálat következő fázisában az összetett mondatok esetét kezelő algoritmusok kidolgozását végezzük el.



6. ábra. Algoritmus időszükséglete

Köszönetnyilvánítás

A cikkben ismertetett kutató munka az EFOP-3.6.1-16-2016-00011 jelű „Fiatallodó és Megújuló Egyetem – Innovatív Tudásváros – a Miskolci Egyetem intelligens szakosodást szolgáló intézményi fejlesztése” projekt részeként – a Széchenyi 2020 keretében – az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Irodalom

- [1] Michael, C, Bangalore, S, Schaeffer, M.: Computational linguistics and translation studies. *Border crossings: Translation studies and other disciplines* 126 (2016), pp. 225-240. <https://doi.org/10.1075/btl.126.11car>
- [2] Krallinger, M. et al.: Information retrieval and text mining technologies for chemistry., *Chemical reviews* 117.12 (2017), pp. 7673-7761. <https://doi.org/10.1021/acs.chemrev.6b00851>
- [3] Spiro, Rand J.: Remembering information from text: The "state of schema approach., *Schooling and the acquisition of knowledge*. Routledge, (2017), pp. 137-165. <https://doi.org/10.4324/9781315271644-11>
- [4] Kovács, L, Barabás, P.: Grammar Representation Forms in Natural Language Interface for Robot Controlling. *Emergent Trends in Robotics and Intelligent Systems*. Springer, Cham, (2015), pp. 65-72. https://doi.org/10.1007/978-3-319-10783-7_7
- [5] de Marneffe, M., Nivre., J. : Dependency grammar. *Annual Review of Linguistics* 5 (2019): pp. 197-218. <https://doi.org/10.1146/annurev-linguistics-011718-011842>
- [6] Jiang, J., Haitao, L.: Lucien Tesnière, Elements of structural syntax. Translated by Timothy Osborne and Sylvain Kahane. Amsterdam & Philadelphia, PA: John Benjamins, 2015. Pp. lxxxii \$+ \$698. *Journal of Linguistics* 51.3 (2015), pp. 705-709. <https://doi.org/10.1017/S0022226715000249>

- [7] Gerdes, K, Kahane, S.: Dependency annotation choices: Assessing theoretical and practical issues of universal dependencies, Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016). (2016) <https://doi.org/10.18653/v1/W16-1715>
- [8] Gruber, T R.: Toward principles for the design of ontologies used for knowledge sharing, International journal of human-computer studies 43.5-6 (1995), pp. 907-928. <https://doi.org/10.1006/ijhc.1995.1081>
- [9] Niles, I, Pease, A: Towards a standard upper ontology. Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001. ACM, (2001) <https://doi.org/10.1145/505168.505170>
- [10] Stojanovic, Ljiljana, et al.: User-driven ontology evolution management, International Conference on Knowledge Engineering and Knowledge Management. Springer, Berlin, Heidelberg, (2002) https://doi.org/10.1007/3-540-45810-7_27
- [11] Xiang, Z, Cheng, G, Qu, Y.: Ontology summarization based on rdf sentence graph., Proceedings of the 16th international conference on World Wide Web. ACM, (2007).
- [12] Grigoris, A, Frank Van Harmelen: Web ontology language: Owl., Handbook on ontologies. Springer, Berlin, Heidelberg, (2004), pp. 67-92. https://doi.org/10.1007/978-3-540-24750-0_4
- [13] Horrocks, Ian.: DAML+OIL: A Description Logic for the Semantic Web. IEEE Data Eng. Bull. 25.1 (2002), pp. 4-9
- [14] Vilho, R, Hyvönen, E.: Ontology-based semantic metadata validation. 46 h Fg E 9 i VIE 7 @ GPQ qhrhrRq@ rh(2002): 6.
- [15] Sowa J.F., Conceptual Structures: Information Processing in Mind and Machine (Addison-Wesley, Reading, MA, USA, (1984)
- [16] Sowa J.F., Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, (2000)
- [17] Creasy P., Moulin B., Extending the Conceptual Graph approach for data conceptual modelling, Data Knowl. Eng., 8, (1992) pp. 223-248. [https://doi.org/10.1016/0169-023X\(92\)90024-6](https://doi.org/10.1016/0169-023X(92)90024-6)
- [18] Faron C., Ganascia J.G., Representation of defaults and exceptions in conceptual graphs formalism, Conceptual Structures: Fulfilling Peirce's Dreams, Lecture Notes Comp. Sc., 1257, (1997), pp. 153-167. <https://doi.org/10.1007/BFb0027868>
- [19] Pfeiffer H.D., Hartley R.T., The Conceptual Programming Environment, CP: Time, Space and Heuristic Constraints, In: Proceedings of the Sixth Annual Workshop on Conceptual Graphs, Binghamton, (1991)
- [20] Soshnikov D., Data and knowledge representation models of distributed frame systems, In: Proceedings of PreConference Workshop of VLDB-2003 Emerging Database Research in Eastern Europe, Brandenburg University of Technology at Cottbus, (2003), pp. 123-127
- [21] Cao T.H., Conceptual Graphs and Fuzzy Logic: A Fusion for Representing and Reasoning with Linguistic Information, Stud. Comput. Intel., 306, (2010)
- [22] Baksa-Varga E., Kovács L., Semantic representation of natural language with Extended Conceptual Graph, J. Prod. Syst. Inf. Eng., 5, (2009), pp. 19-39.