Bevezetés az `e-magyar` programcsomag használatába

Mittelholcz Iván
fejlesztőmérnök
MTA Nyelvtudományi Intézet
mittelholcz.ivan@nytud.mta.hu

2006 óta dolgozom az MTA Nyelvtudományi Intézetében szoftverfejlesztőként. Érdeklődési körömbe olyan nyelvtechnológiai feladatok tartoznak, mint a természetes nyelvi szövegek tokenizálása, a helyesírás-ellenőrzés vagy az ontológiaépítés. Foglalkozom felügyelt gépi tanulással, webprogramozással és projektek számára kényelmesen használható számítógépes infrastruktúra kialakításával is.

1. Mire jó?

A számítógépes nyelvészet természetes nyelvi szövegek automatikus elemzésével vagy gépi feldolgozásával foglalkozik, ezt a célt szolgáló szoftvereket fejleszt. A diszciplína története során kialakult a szövegfeldolgozási lépéseknek egy többé-kevésbé egymásra épülő lánca. Ez olyan lépéseket foglal magában, mint a szöveg felbontása mondatokra és szavakra, a szavak morfológiai elemzése, a szófaj megállapítása, a mondatok szintaktikai elemzése, továbbá bizonyos sajátos mondatalkotó kifejezések felismerése a szövegben (pl. tulajdonnevek).

Az `e-magyar` programcsomag is egy ilyen, a fenti elemzési pontokat megvalósító eszközlánc. A lánc egymásra utalt modulokból áll. Egy modul a láncban előtte lévő modul kimenetén kezd dolgozni, ahhoz hozzáteszi a maga elemzéseit, majd továbbadja az egészet a következő modulnak. Egy ilyen láncba az első modul a még teljesen elemzetlen szövegen kezd el dolgozni, míg az utolsó kimenete a már teljesen elemzett szöveg, amit már nem fog további modul feldolgozni. Az informatikában pipeline-nak, azaz csővezetéknek szokták ezt az architektúrát nevezni. Ahogy a fizikai csővezetékben – gondoljunk itt, mondjuk, a kőolajfinomításra – halad a kezdetben nyers olaj, és válik a feldolgozás során lépésről lépésre finomabbá[5], úgy halad át a szöveg is egy ilyen elemzőláncon, és válik fokozatosan egyre feldolgozottabbá.

[5] Hogy finom nem lesz, azt már Besenyő Istvántól is tudhatjuk.

A szöveg feldolgozottsága az úgynevezett annotációkban vagy címkékben ölt testet. Az egyes elemzőmodulok az általuk megállapított információkat ilyen címkékben fűzik hozzá a szöveghez. Az annotációnak vagy címkézésnek a formátuma meglehetősen sokféle lehet. Például egy XML-alapú jelölésben így nézhet ki egy szófaji címke: `<szó szófaj="főnév">alma</szó>`.

Az `e-magyar` elemzőlánc a főbb magyarországi nyelvtechnológiai műhelyek összefogásával készült.[6] A modulok többnyire nem a nulláról indultak, hanem a műhelyek már meglévő programjai lettek átdolgozva, összehangolva. Magyar nyelvre eddig egy hasonlóan komplett elemzőlánc készült csak, mégpedig a szegedi magyarlánc.[7] A magyarlánc egyes komponensei az `e-magyar`nak is részei.

Az `e-magyar` programcsomag a kezdetektől úgy lett tervezve, hogy mind a szakmabeli nyelvtechnológusok és fejlesztők, mind az érdeklődő nagyközönség hasznát tudja venni. A szakma számára az `e-magyar` nyílt forráskódú projektként elérhető és Linux operációs rendszerekre telepíthető a projekt GitHub oldaláról.[8] A nagyközönség számára az `e-magyar` honlapján keresztül elérhető az elemzőlánc egy online kipróbálható változata. A következőkben ezt mutatjuk be.

2. Hogyan működik?

Az online változat a http://e-magyar.hu/hu/parser címen érhető el. Bárki számára ingyen használható, viszont az egyszerre elemezhető szövegek terjedelme 6000 karakterben van korlátozva. Akinek ennél nagyobb igényei vannak, az kénytelen telepíteni a programcsomagot és a saját számítógépén elvégezni az elemzést.

A nyelvtechnológiában a feladatok megoldásának két alapvető módszere van. Az első csoportba a szabályalapú megközelítések tartoznak. Ezek közös jellemzője, hogy `ha …, akkor …` típusú szabályokból álló feltételrendszerekkel kezelik a problémákat.

A második csoportba az úgynevezett statisztikai megközelítések tartoznak. Ezekre az igaz, hogy nincsenek általános, kőbe vésett szabályok, hanem sok, már

[6] MTA Nyelvtudományi Intézet, Pázmány Péter Katolikus Egyetem, Szegedi Tudományegyetem, MTA SZTAKI, AITIA International Zrt., Morphologic Kft.
[7] https://rgai.inf.u-szeged.hu/node/100
[8] https://github.com/dlt-rilmta/e-magyar-tsv. Megjegyzendő, hogy a projekt most egy nagyobb átdolgozáson esik át. Ez az új változat URL-e. A régebbi változat elérhetősége: https://github.com/dlt-rilmta/hunlp-GATE.

megvizsgált adat alapján állít fel az elemző egy statisztikai modellt, és az elemzés során ez a modell mondja meg, hogy milyen címkével kell ellátni az elemzendő nyelvi kifejezést.

Az `e-magyar` programcsomag egyes elemzői szabályalapúak, míg mások statisztikaiak. Hogy mikor és melyik módszert érdemes használni, az a problémán, az elemzési feladaton múlik. A szabályalapú módszer gyors és könnyen tud kezelni egyszerűbb, jól körülhatárolható feladatokat, de könnyen válik emberileg átláthatatlanná, nehezen javíthatóvá, ha egyszerre sok feltételt, alesetet kell kezelni. Megszokott tapasztalat az elbonyolódó szabályrendszereknél, hogy egy dolog kijavításával több új hibát „vezet be" az ember.

Ilyen esetekben lehet hasznos a statisztikai megközelítés,[9] ami jól kezel sok, akár logikátlan, kivételeket tartalmazó esetet is egyszerre. Azonban ennek is megvan a maga hátránya: sokszor a programok fejlesztői vagy használói maguk sem tudják, hogy miért úgy működik az elemzőjük úgy, ahogy.

A most következő alfejezetekben röviden azt ismertetjük, hogy az elemző egyes moduljai mit csinálnak, és azt hogyan teszik.

2.1. Mondatra bontás, tokenizálás

Az első modul két dolgot is csinál egyszerre: a nyers szöveget először mondatokra bontja, aztán a mondatokat úgynevezett tokenekre.

A mondatra bontásnak kettős szerepe van. Egyrészt vannak olyan elemzők a láncban, amelyek mondatokon dolgoznak, nem csak szavakon (ilyenek például a szintaktikai elemzők), másrészt a szöveg szavakra bontása is könnyebbé válik, ha már megvannak a mondathatárok. A mondatra bontás egyszerű feladatnak tűnhet (és igazából az is), de van pár nehézség benne. Ilyen például a rövidítések megfelelő kezelése. A következő példamondatban például kifejezetten hiba a pont és nagybetű között mondathatárt sejteni.

Támogatta a haladó eszméket, barátságban állt pl. Jókai Mórral is.

A tokenizálás feladata a mondathatárok között megkeresni az értelmes, a morfológiai elemzőnek továbbadható szavakat. El kell különíteni a szóközöket, az

[9] Ezt szokták gépi tanulás névvel is illetni, abból a hasonlatból kiindulva, hogy ha sok adatot mutatunk egy gépnek (igazából egy programnak), akkor az olyan, mintha „tanítanánk".

írásjeleket, megfelelően kell kezelni a dátumokat, mértékegységeket, az olyan informatikai kifejezéseket, mint e-mail-címek, URL-ek stb.

Mivel a mondatra bontás és a tokenizálás is viszonylag egyszerű feladatnak számít, ezért az `e-magyar`ba is szabályalapon működő program került.

2.2. Morfológiai elemzés

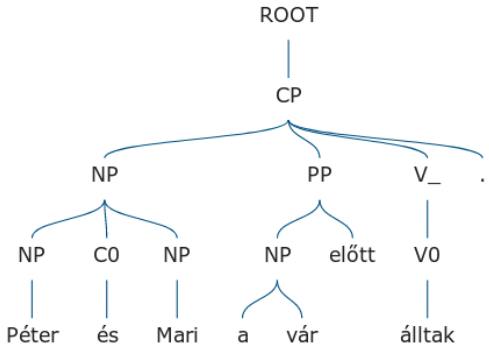
A következő elemzési szint a morfológiai elemzés. A tokenizálóból kijövő szavakat a morfológiai elemző egyenként beolvassa, és minden egyes szóhoz hozzárendeli az összes lehetséges morfológiai elemzését. A morfológiai elemzés tartalmazza a szóalak felbontását toldalékokra, képzőkre és összetételi határokra. Az elemző nem veszi figyelembe a szavak kontextusát, nem figyel arra, hogy az adott szó a mondat elején vagy végén található-e, hogy milyen más szavak vannak a környezetében stb. Csak magát a szóalakot veszi figyelembe, és próbálja azt minden lehetséges módon felbontani. A morfológiai elemzés ezért sokszor hoz meglepő eredményeket. Így lesz például a fenti példamondat *haladó* szavának öt lehetséges elemzése, köztük olyanok, mint a *hal* + *adó,* azaz 'halakra kivetett adó', vagy a *hal* + *ad* + *ó,* azaz 'halakat adományozó'.

Az `e-magyar`ban lévő morfológiai elemző is szabályalapon működik, konkrétan egy véges állapotú transzdúcernek nevezett technológia van mögötte.

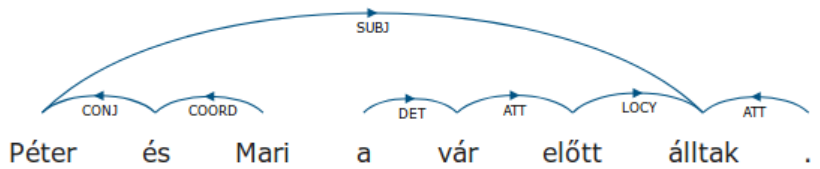
A morfológiai elemzőhöz szorosan kapcsolódik a szótövesítő modul. Ennek feladata, hogy a teljes morfológiai elemzésből előállítsa az elemzett szó szótári alakját és szófaját.

2.3. Morfológiai egyértelműsítés

A morfológiai elemzés kimenete tartalmazza minden szó minden lehetséges elemzését. Az egyértelműsítés feladata, hogy megállapítsa, a lehetséges elemzésekből melyik a helyes az adott mondatban. Például a „*Péter Marira vár*." mondatban a *vár* egyes szám harmadik személyű, jelen idejű ige, míg a „*Péter a vár előtt áll*." mondatban a *vár* alanyesetű főnév.



1. ábra: Összetevős elemzés



2. ábra: Függőségi elemzés

Ahhoz, hogy a sok lehetséges elemzés közül ki lehessen választani a megfelelőt, már nem elég a szavakat magukban nézni. Az egyértelműsítő modul már mondatokat vizsgál, és a lehetséges elemzések legvalószínűbb láncát keresi.

Az egyértelműsítés már nem szabályalapú, hanem statisztikai módszereken nyugszik. A módszer lényege, hogy sok, már elemzett és emberek által egyértelműsített mondatot mutatunk a programnak. Az egyértelműsítő ez alapján megtanulja, melyek a tipikus vagy gyakoribb mintázatok egy szövegben, és melyek a valószínűtlenek. Nem egyszerűen azt nézi, hogy a *vár* gyakrabban ige, mint főnév. Számára az a fontos, hogy egy névelőt sokkal gyakrabban követ főnév, mint ige, ezért az „*a vár*” kifejezésben a *vár*-at főnévként egyértelműsíti. Hasonlóan, a *Marira vár* esetében a szublatívuszi esetragú főnév után valószínűbb az ige, mint a főnév, ezért az igei címkét fogja kapni az elemzőtől.

Ehhez hasonló, feltételes valószínűségeken alapuló számításokkal tudja a program megállapítani a legvalószínűbb címkesorozatot minden mondatnál.

2.4. Szintaktikai elemzés

A szintaktikai elemzés, az egyértelműsítéshez hasonlóan, szintén mondatokon működik, és morfológiailag már egyértelműsített szöveg mondatainak építi fel a szintaktikai fáját. Az `e-magyar` programcsomag kétféle szintaktikai elemzőt tartalmaz.

A Chomsky-féle generatív nyelvtanon alapuló összetevős elemzés a mondatot kisebb kifejezésekre bontja (főnévi, igei stb. frázisokra), egészen addig, amíg el nem jut a mondatot alkotó szavakig. A fa leveleit a mondat szavai (terminálisok) alkotják, a fa nem-levél csomópontjai pedig az egyre összetettebb kifejezések (nem-terminálisok). A fában lévő élek címkézetlenek (lásd az 1. ábrát).

A függőségi elemzés olyan elemzési fát ad kimenetül, ami a szavak közötti relációt fejezi ki. A fa minden csomópontja egy szó, minden csomópontok közötti él a gyerek (függő) csomóponttól mutat a szülő csomópont felé. A függőségi fa élei a függőségi viszony típusával vannak címkézve (lásd a 2. ábrát).

Mindkét szintaktikai elemző statisztikai módszerrel dolgozik, és a már említett magyarláncból lett átvéve. Az összetevős elemző a Berkeley parser,[10] a függőségi elemző pedig a Bohnet parser[11] magyarra igazított változata.

2.5. Főnévi csoportok és tulajdonnevek felismerése

A következő két elemző nagyon hasonló elven működik. Az egyik maximális főnévi kifejezéseket keres a szövegben,[12] a másik a tulajdonneveket igyekszik felismerni. Mind a két feladat az információkinyerésnek nevezett problémához jelent segítséget. Az információkinyerés célja, hogy természetes nyelvi szövegekből olyan alapvető információkat nyerjen ki, mint hogy ki, kivel, mikor és mit csinált. Ezek az információk alapvetően NP-k vagy tulajdonnevek által jelölt entitások közötti relációkat írnak le. Ahhoz, hogy az entitások közötti relációkat sikerüljön egy szövegből automatikusan kinyerni, előbb fel kell ismerni magukat a szövegben szereplő entitásokat. Ezt a célt szolgálja ez a két modul.

[10] https://github.com/slavpetrov/berkeleyparser
[11] https://code.google.com/archive/p/mate-tools
[12] Maximális főnévi kifejezéseknek vagy NP-knek azokat az NP-ket nevezzük, melyek nem részei egy nagyobb főnévi kifejezésnek sem.

Mindkettő ugyanazon a gépi tanulási módszeren alapul.[13] Ennek lényege a szokásos gépi tanulásos eljáráson alapul: sok olyan szöveget mutattunk a programnak, amiben már be vannak jelölve a tulajdonnevek vagy a maximális NP-k, és megmondjuk azt is a programnak, hogy a szövegben a szavak milyen egyéb tulajdonságaira figyeljen. Például: hogy kis- vagy nagybetűvel kezdődik-e, hogy mondat elején van-e, hogy milyen a szófaja stb. A program megpróbált összefüggéseket találni ezen tulajdonságok és a címkék között, és előállított egy ún. valószínűségi modellt. Az `e-magyar`ba beépítettük ezt a valószínűségi modellt, amit az elemzőmodul arra használ, hogy a címkézetlen szövegben a szavak tulajdonságai alapján megpróbálja a címkéket megállapítani.

3. Mik a korlátai?

A nyelvtechnológiai rendszerek nem tökéletesek, ez sok feladat esetében elvileg is lehetetlen. A cél általában nem is egy tökéletes rendszer elkészítése, hanem egy „elég jó” rendszeré.

A hasonló feladatokra készült elemzőket két fő paraméter alapján szoktuk összehasonlítani. Az elsőt pontosságnak (precision) nevezik, és ez annak arányát jelenti, hogy a kiosztott címkékből mennyi a helyes. Például, ha az elemzendő szövegben száz kifejezést jelölt tulajdonnévként a program, és ebből nyolcvan volt helyes, akkor a rendszerünk pontossága 0,8. A másik mérőszámot fedésnek (recall) nevezzük, és ez az eltalált és a szövegben ténylegesen meglévő dolgok arányát jelenti. Például, ha a szövegben száz maximális főnévi kifejezés volt, és a rendszerünk ebből hetvenet talált el, akkor a fedés 0,7 lesz.

Az elemzők ilyenfajta kiértékelése feltételezi, hogy van olyan szövegünk, amelyet emberi munkával is elemeztünk, és ezt az elemzést biztosnak fogadjuk el, hogy összehasonlíthassuk a gépi elemző kimenetével.

4. Ajánlott irodalom

Az `e-magyar` eszközláncról négy cikk is megjelent a 2017-es Magyar Számítógépes Nyelvészeti Konferencia kiadványában.[14]

[13] Mindkettő a HunTag3 programot használja, lásd https://github.com/ppke-nlpg/HunTag3

[14] Elérhető a https://rgai.inf.u-szeged.hu/sites/rgai.sed.hu/files/kotet.pdf címen.

1. Általános áttekintést nyújt az `e-magyar` programcsomag egészéről:

Váradi T., Simon E., Sass B., Gerőcs M., Mittelholcz I., Novák A., Indig B., Prószéky G., Farkas R., Vincze. V. 2017. Az `e-magyar` digitális nyelvfeldolgozó rendszer. In: Vincze V. (szerk.). *XIII. Magyar Számítógépes Nyelvészeti Konferencia*. *MSZNY 2017*. Szeged: Szegedi Tudományegyetem, Informatikai Intézet. 49–60.

2. Egy cikk az `e-magyar` tokenizáló moduljáról:

Mittelholcz I. 2017. emToken: Unicode-képes tokenizáló magyar nyelvre. In: Vincze V. (szerk.). *XIII. Magyar Számítógépes Nyelvészeti Konferencia*. *MSZNY 2017*. Szeged: Szegedi Tudományegyetem, Informatikai Intézet. 61–69.

3. A morfológiai elemző alapos bemutatása:

Novák A., Rebrus P., Ludányi Zs. 2017. Az emMorf morfológiai elemző annotációs formalizmusa. In: Vincze V. (szerk.). *XIII. Magyar Számítógépes Nyelvészeti Konferencia*. *MSZNY 2017*. Szeged: Szegedi Tudományegyetem, Informatikai Intézet. 70–78.

4. Az `e-magyar` GATE-integrációjáról szól, de a bennefoglaltak már nem feltétlenül helytállók. Az `e-magyar` jelentős átdolgozáson esett át mostanában, aminek egyik fontos része pont a GATE keretrendszertől való megszabadulás volt:

Sass B., Miháltz M., Kundráth P. Az e-magyar rendszer GATE környezetbe integrált magyar szövegfeldolgozó eszközlánca. In: Vincze V. (szerk.). *XIII. Magyar Számítógépes Nyelvészeti Konferencia*. *MSZNY 2017*. Szeged: Szegedi Tudományegyetem, Informatikai Intézet. 79–90.

A kötetben található további két cikk, melyek szintén az `e-magyar` projekt keretében készült eszközökről szólnak, de ezek nem képezik szerves részét az ismertetett eszközláncnak.

A magyar nyelv és nyelvtechnológia helyzetét ismerteti:

Simon E., Lendvai P., Németh G., Olaszy G., Vicsi K. 2012. *A magyar nyelv a digitális korban. The Hungarian Language in the Digital Age*. Springer.

A http://www.meta-net.eu/whitepapers/e-book/hungarian.pdf címen letölthető könyv külön figyelmet fordít arra, hogy a jelen kor kihívásainak fényében tárgyalja a nyelvészeti és nyelvtechnológiai témákat.

Gyakorlatorientált bevezetést nyújt Mittelholcz Iván és Simon Eszter közös számítógépes nyelvészeti kurzusának anyaga. Ez a https://github.com/m-ivan/compling címen érhető el.

Ennél alaposabb és elméletibb bevezetést ad a nyelvtechnológia problémáiba és módszereibe:

Jurafsky, D., Martin, J. H. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River: Prentice Hall.

A klasszikus könyv készülő, harmadik kiadása online elérhető a https://web.stanford.edu/~jurafsky/slp3 címen.