PARANCSKINYERÉS MAGYAR NYELVŰ SZÖVEGBŐL

FUNCTION EXTRACTION FROM HUNGARIAN TEXT

Barabás Péter[*]

ABSTRACT

In human-computer interaction the natural language processing is an important area. Communicating with computer generally is quite difficult in natural language, thus processing sentences will be limited to a specific domain to be able to build knowledge base up and to evaluate questions and commands in tolerable time. The goal of our research is to develop a natural language interface framework with help of which applications and systems can be controlled with Hungarian language.

1. BEVEZETÉS

A számítógépek megjelenése magával hozta bizonyos mesterséges nyelvek kialakulását, amelyek segítségével kommunikálni tudunk a gépi rendszerekkel. Ezen nyelvek tipikus példái a programozási nyelvek. Az emberek a számítógépek megjelenése óta szeretnék megvalósítani, hogy a gépekkel az egymás között megszokott, természetes nyelven „beszélgessenek". Ezen interakciónak a gyakorlati megvalósulását a természetes nyelvi felületek jelentik. A természetes nyelvi interfésszel (NLI) rendelkező információs rendszerek gyökerei 1970-re nyúlnak vissza. Az úttörő LUNAR [1] projekt a holdkőzetek adatbázisában való lekérdezésekhez dolgozott ki természetes nyelvű interfész felületet. A RENDEZVOUS (Codd, 1977) rendszer volt az első általános célú adatbázis NLI modul. Az NLI modulok egyik alapfeladata a természetes nyelven beérkező parancsok átkonvertálása a feldolgozó modul saját parancsnyelvére Ezen konverzió megvalósítása több lépcsőben történik kezdve a természetes nyelvi mondat szintaktikai ellenőrzésével, elemzésével és folytatva a szemantikai analízissel, a tématerület ismert fogalmainak detektálásával.

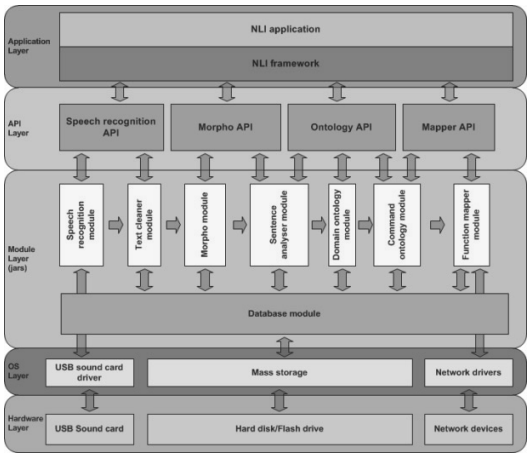
Több ismert NLP keretrendszer is található a piacon, ezek közül a legismertebbek az Apache nyílt forráskódú OpenNLP [3] és UIMA [4] rendszerei, illetve a Stanford Egyetem statisztikai alapú Stanford NLP [5] rendszere. Az előzőekben felsorolt rendszerekben egyaránt megtalálhatóak a legfontosabb szöveg feldolgozási modulok, mint a mondat- és szódetektáló, tulajdonnév azonosító, szótövező, mondat osztályozó, stb. A legtöbb létező keretrendszer alapvetően az angol nyelvet, illetve még esetleg néhány „könnyebben" feldolgozható nyelvet támogat. Esetenként előfordul, mint az UIMA esetén is, hogy a keretrendszer adaptálható különböző nyelvekre.

A magyar nyelv egy nehezen feldolgozható nyelv, amely az agglutináló tulajdonsága révén szavak és toldalékok nagyszámú kombinációjával rendelkezik, melyek között sok szabályszerűség és kivételes ragozás fedezhető fel. Magyar nyelvű kutatásokban [6][7] is találkozhatunk az előzőkben említett rendszerek használatával.

A kutatásunk célja, hogy az előzőekben említett NLP keretrendszerek egyes funkcióit magyar nyelvű szöveg feldolgozására implementáljuk és kiterjesszük a parancsnyelvi funkciókra történő konvertálással és azok implementációjának dinamikus meghívásával.

2. NLP RENDSZER STRUKTÚRÁJA

A szöveges input elemzése egy meglehetősen bonyolult és összetett folyamat, mely alfolyamatok, almodulok láncolataként kezelve oldható meg



1. ábra. NLP rendszer logikai struktúrája

[*] *egyetemi tanársegéd, Miskolci Egyetem, ÁIT*

hatékonyan. Az 1. ábrán látható, hogy az NLI rendszer a következő almodulokra bontható:

- Bemeneti eszközhöz kapcsolódó modulok
  - Beszédfelismerő modul
  - Szövegtisztító, átalakító modul
- Morfológiai modul
- Fogalmak feldolgozásához kapcsolódó modulok
  - Domain ontológiai modul
  - Mondatelemző modul
- Parancsfeldolgozás moduljai
  - Parancsontológiai modul
  - Funkciókinyerő modul

A modulok két részre oszthatók a nyelvfüggőség tekintetében:

- Nyelvfüggő modulok: HID modulok és a morfológiai modul
- Nyelvfüggetlen v. kvázi-nyelvfüggetlen modulok: ontológiai modulok (domain, parancs)

2.1. Szövegtisztító modul

A szövegtisztító modulnak képesnek kell lennie a bemeneti szöveg hibáinak detektálására, javítására. Mivel a bemenet származhat többféle forrásból is (beszéd, billentyűzet, lapolvasó, kézírás), az analóg-digitális konverterek a különböző források esetén más és más hibákat ejthetnek. A billentyűzetről származó szövegben az elütések okozhatnak hibákat, melyek javításánál a billentyűk elhelyezkedését tudjuk figyelembe venni, míg egy scannerből származó forrásban a hasonló formájú karakterek lehetnek hibásan felismerve. Ezen hibák javítására különböző algoritmusok használhatóak a hibaforrás függvényében.

Ahhoz azonban, hogy a további modulok számára használható bemenetet biztosítsunk, a modul funkciói közé tartoznak a következők:

- Szótárkezelés
- Mondathatárolók és szóhatárolók kezelése
- Mondatok detektálása
- Szavak detektálása
- Szavak helyesírás elemzése, javítása

2.2. Morfológiai modul

A morfológiai modul szerepe a mondat(ok) szavainak nyelvtani elemzése. A további modulok feladata a mondatok jelentésének meghatározása, melyhet elengedhetetlen a szavak morfológiai analízise, ugyanis ugyanaz a szótő más-más toldalékkal teljesen más jelentést hordozhat. A morfológiai analízisen túl szükség lehet egy tetszőleges szó adott szabály szerinti ragozott alakjának meghatározására is. Ez utóbbi funkció a válaszgenerálásnál kaphat jelentősebb szerepet.

A morfológiai modulnak a következő funkcionalitással kell rendelkeznie:

- Szótövek tárolása, kezelése
- Nyelvtani szabályok tárolása, kezelése
- Kivételesen ragozandó szavak kezelése
- Szavak morfológiai analízise
- Szavak ragozása
- Analízis pontosságának meghatározása adott teszt szóhalmazra

2.3. Domain ontológiai modul

A domain ontológiai modul kezeli a jelentését a szövegnek felhasználva ez előző modulok eredményeit. A jelentés meghatározásának alapegységei a fogalomak, melyek között igen változtos kapcsolatrendszer építhető ki. Egy adott domain fogalmait, fogalmi kapcsolódásait lehet ezen modul funkcióival felépíteni.

Az egyes fogalmak írásbeli, illetve beszédbeli megvalósulásai a szavak, melyek nyelvenként eltérőek, azonban a jelentésükről elmondható, hogy többé-kevésbé nyelvfüggetlenek. Például amikor meghalljuk azt a szót, hogy telefon, mindenki egy készülékre gondol, amellyen keresztül távol lévő emberek tudnak egymással beszélgetni, függetlenül attól, hogy ezt egy magyar ember teszi, aki “telefon”-nak nevezi a készüléket vagy egy angol ember teszi, aki “phone”-nak vagy “telephone”-nak.

A domain ontológiai modul feladatai közé tartoznak a következők:

- Fogalmak kezelése
- Fogalmak detektálása a mondatból
- Mondat fogalmi kapcsolatainak meghatározása

2.4. Mondatelemző modul

A mondatelemző modul a morfológiai modul kimenetét használja ahhoz, hogy felépítse a bemenet mondattani elemzését. Ahhoz, hogy a mondat jelentését megértsük, szükség van arra, hogy az egyes szavakról, szókapcsolatokról meg tudjuk mondani, hogy milyen mondatrészt töltenek be a szövegben. Itt kerülnek meghatározásra az egyes mondatrészek: alany, állítmány, tárgy, stb., és a mondatok közötti viszonyok: alá-, mellérendeltség. A mondatelemzés nem végezhető el tisztán szintaktikai úton ismerve az egyes szavak morfológiai analízisét, ugyanis nagyon sok esetben a

predikátumból, illetve annak jelentéséből tudjuk, hogy egy adott szó milyen mondatrészi szereppel bír a mondatban.

A mondatelemző modulnak a következő funkciókat kell megvalósítania:

- Mondatelemző tanítása
- Mellékmondatok meghatározása, detektálása
- Mondatrészek meghatározása

2.5. Parancsontológiai modul

A parancsontológiai modul az NLI keretrendszer utolsó modulja, melynek feladata a kiadott parancs, utasítás végrehajtása. Egy domain-orientált alkalmazásban véges számú utasítást tudunk kezelni. A parancsok leírásának általános, absztrakt módon kell történnie, hogy tetszőleges alkalmazást, eszközt ki lehessen terjeszteni a természetes nyelvi irányítással.

A végrehajtandó funkciók közös tulajdonsága, hogy van neve, amely azonosítja őket, illetve nulla vagy több paraméterrel rendelkeznek. A feldolgozás során a természetes nyelvi mondatokban detektált fogalmakat kell tudni megfeleltetni a parancsleírásban szereplő elemeknek, majd a kiválasztott („nyertes”) funkciót a detektált paraméterekkel futtatni és az eredményt megjeleníteni.

A parancsontológiai modul fő feladatait a következőképpen foglalhatjuk össze:

- Funkciók kezelése
- Funkcióosztályozó kezelése, tanítása
- Funkciók, paraméterek fogalmakhoz társítása
- Kérdésgenerálás hiányzó paraméterekhez
- Parancs végrehajtás
- Eredménygenerálás

A keretrendszer legfontosabb feladata a mondatokban lévő fogalmak detektálása és azok társítása a funkciókhoz. Erről a következő fejezetben lesz szó részletesen.

3. FOGALOMKINYERÉS TERMÉSZETES SZÖVEGBŐL

A szövegtisztító és morfológiai modulok pontos működése nagymértékben hozzájárul a szövegfeldolgozás sikerességéhez. Ezen modulok a szintaktikai feldolgozást végzik, éppen ezért elkülönülnek a további moduloktól, amelyek feladata a szöveg „jelentéstartalmának” elemzése, kezelése.

Ahhoz, hogy a mondatok fogalmait meghatározhassuk, szükségünk van a domaint alkotó fogalomháló előzetes, tudásbázisként történő definiálására. A fogalmak leírására használhatunk létező nyelveket, mint az OWL [8] vagy RDF [9], de megtehetjük ezt saját leíró struktúrák definiálásával egyaránt. A fogalmak között a következő típusú kapcsolatok definiálhatók:

- Leszármazás kapcsolat: a gyerek elem a szülőnek egy leszármazottja, egy specifikus példánya.
- Predikátum kapcsolat: a szülő elem egy predikátum, a mondatban az állítmány szerepét tölti be, a gyerek elem pedig egy másik, nem állítmány mondatrészhez tartozó fogalom
- Tulajdonság kapcsolat: a szülő egy tetszőleges fogalom, a gyerek elem pedig az ahhoz kapcsolódó tulajdonságot, többnyire jelzőt meghatározó fogalom

Egy fogalomnak több, különböző írásbeli reprezentációja is lehetséges. Ezeket nevezhetjük szinonimáknak is, azonban ezek hatásköre a domainre korlátozódik. Más szövegkörnyezetben már teljesen más jelentést hordozhat ugyanaz a szó. Ezért is fontos, hogy pontosan meghatározzuk, hogy egy adott szó melyik domain fogalmához is tartozik.

A fogalom kinyerés során egy a bejövő mondathoz rendelt fogalomhalmazt kapunk, amelyben alapvetően nincs sorrendiség. Ezzel a magyar nyelv szabad szórendje is egyszerűen kezelhetővé válik. Természetesen a kapcsolatoknál, tipikusan a tulajdonság kapcsolatoknál fontos lehet a megelőzési sorrend, amely a szavak mondatbeli indexéből levezethető.

A funkciótársításhoz a detektált fogalmak mondatbeli szerepét is ismernünk kell, ugyanis egy szó különböző szövegkörnyezetben különböző jelentéssel, illetve mondatbeli szereppel bírhat. A mondatelemzést a következő lépésekben végezhetjük:

1. A predikátum meghatározása. Amennyiben nincs, a „van” létige használata.
2. A predikátum fogalom predikátum kapcsolatainak lekérdezése.
3. A predikátumhoz kapcsolható fogalmak illesztése a mondatban detektált fogalmakhoz.
    a. A fogalom illesztése.
    b. A toldalékok illesztése.
4. A detektált mondatrészek visszaadása a további modulok által feldolgozható formában *(XML)*.

A mondatelemzés eredményként tartalmazza a bejövő mondat fogalmait és azok mondatbeli szerepét. Ezeket az információkat felhasználva már lehetségessé válik a mondat a megfelelően definiált funkciókhoz való társítása.

4. FUNKCIÓTÁRSÍTÁS FOGALMAKHOZ

Az előzőekben kinyert információt felhasználva utolsó lépésben a megfelelő funkciót kell kiválasztanunk. A funkciók leírására a következő jelölésrendszert vezethetjük be:

$$F = (Pred, Par, Im),$$

ahol
$Pred = \{pred_i\}, i = 1, \dots, N$: *a predikátumok,*
$Par = \{par_j\}, j = 0, \dots, M$: *a paraméterek,*
$Im = \{Class, Method\}$: *az implementáció*

Egy funkcióhoz több predikátum fogalom is társítható, legalább egy azonban kötelező. A mondatban detektált állítmány fogalmát kell a funkció kiválasztásnál az egyes funkciók predikátum fogalmaihoz társítani.

A predikátum alapján illeszkedő funkciók implementációjának meghívásához a paraméterek megfelelő társítása elengedhetetlen. A paramétereket a következőképpen írhatjuk le:

$$par_i = (n, r, t, C, PS),$$

ahol
n: *a paraméter neve,*
$r = \{true, false\}$: *a paraméter kötelezősége,*
$t = \{ string, int, float, bool\}$: *a paraméter típusa,*
$C = \{c_i\}, i = 1, \dots, N$: *a kapcsolódó fogalmak,*
$PS = \{ps_j\}, j = 1, \dots, M$: *a mondatbeli szerepek*

A mondat még fel nem dolgozott fogalmait illesztenünk kell az egyes paraméterek fogalomlistájára. Egy funkció esetén több paraméterhez is köthetjük ugyanazt a fogalmat, így a pontosítás végett az illesztésnél figyelembe kell vennünk a mondatbeli szerepet (*PS*) is.

Előfordulhat olyan eset is a természetes mondatban, hogy több ugyanazon fogalommal és mondatbeli szereppel bíró mondatrészünk van, pl. több jelző a mondatban különböző szavakhoz. Ezen fogalmak paraméterekhez társítása egy megelőzési, vagy függőségi leíró bevezetését kívánja meg a következő formában:

$$ps_i = \{psn, D\},$$

ahol
psn: *a mondatbeli szerep neve,*
$D = \{d_k\}, k = 0, \dots N$: *a rákövetkező mondatrészek*

Így egyező mondatrészek esetén a kapcsolódó, azaz rákövetkező fogalom mondarésze dönti el, hogy melyik paraméterhez kell a fogalmat társítani.

A paraméterek társítása után azon funkció kerül kiválasztásra, amelynél egyrészt predikátumilleszkedés van, másrészt a legtöbb „kötelező” paraméter lefedhető a mondatban kinyert fogalmakkal. Amennyiben valamely „kötelező” paraméterhez nem társítható fogalom a mondatból, a rendszer visszakérdezéssel próbálja kikényszeríteni a hiányzó paraméter megadását. Amikor a funkció kiértékelhetővé válik, a hozzá tartozó implementáció (*Im*) kerül meghívásra elérve ez által a kitűzött célt.

5. EREDMÉNYEK

A kutatásaink során sikerült három mintaalkalmazást készíteni eltérő domainekre (navigációs alkalmazás, árfolyam-lekérdező alkalmazás, robotvezérlés), igazolva ezzel a módszer működőképességét. A megvalósított rendszer még összetett mondatokat nem tud kezelni, ez a következő kutatásokban kerül kidolgozásra és megvalósításra.

A bemutatott kutató munka a TÁMOP-4.2.1.B-10/2/KONV-0001-2010 jelű projekt részeként az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

6. IRODALOM

[1] WOODS, W., KAPLAN, R.: Lunar rocks in natural English: Explorations in natural language question answering, Linguistic Structures Processing. In Fundamental Studies in Computer Science, 5, pp. 521-569, 1977.
[2] E.F. CODD.: Seven steps to rendezvous with the casual user. In IFIP Working Conference Data Base Management, 179–200, 1974.
[3] http://opennlp.apache.org
[4] http://uima.apache.org
[5] http://nlp.stanford.edu/software/index.shtml
[6] Zsibrita, János; Nagy, István; Farkas, Richárd 2009: Magyar nyelvi elemző modulok az UIMA keretrendszerhez. In: Tanács Attila, Szauter Dóra, Vincze Veronika (eds.): VI. Magyar Számítógépes Nyelvészeti Konferencia. Szeged, Szegedi Tudományegyetem, pp. 394-395.
[7] Zsibrita, János; Vincze, Veronika; Farkas, Richárd 2010: Ismeretlen kifejezések és a szófaji egyértelműsítés. In: Tanács Attila, Vincze Veronika (eds.): VII. Magyar Számítógépes Nyelvészeti Konferencia. Szeged, Szegedi Tudományegyetem, pp. 275-283.
[8] http://www.w3.org/TR/owl-features/
[9] http://www.w3.org/RDF/