FORMÁLIS NYELVTANI MODUL A META KERET-RENDSZER SZÁMÁRA

FORMAL GRAMMAR MODUL IN GI META FRAMEWORK

Tóth Zsolt[*]

ABSTRACT

There are huge amount of knowledge encoded in written format in a sort of natural language. Hence text processing and information extraction are becoming more and more important fields of computer science and knowledge engineering these days. These processes can be improved by knowledge about the structure of grammar of the language. Theory of formal languages and grammars give the mathematical background to model natural and artificial languages. We are interested in the automatic induction of formal grammars, especially the induction of context free grammars. In this paper the formal grammar module of META[1] framework, which is a grammar induction (GI) and text mining framework, is presented. The presented framework and its module will be used in our further research on context free grammar induction methods.

1. BEVEZETÉS

Napjainkban a tárolt információ jelentős részét természetes nyelven, írott szöveges dokumentumok például levelek, jelentések, elemzések, könyvek formájában tárolják. A korszerű információkereső rendszerek célja a dokumentumok feldolgozása, kivonatolása, tárolása, indexelése a gyors és hatékony keresés céljából. A dokumentum feldolgozás hatékonyságának javítására egy elterjedt módszer, hogy a feldolgozó motort információval látják el az adott hordozó nyelv struktúrájáról. A nyelvtani szabályok ismeretében a feldolgozó motor hatékonyabban képes kinyerni a hordozott információt, így javítva a rendszer pontosságát, működési paramétereit.

Az egyes nyelvtanok matematikai leírására a formális nyelvek szolgálnak, ahol a nyelvet az azt leíró nyelvtan segítségével lehet megadni. A nyelvtan manuális előállítása szakértelmet igénylő és időigényes feladat. Amíg a nyelvtanok automatikus generálása NP nehéz probléma. Ennek az NP nehéz problémának a megoldására különböző megközelítésű, hatékonyságú módszerek léteznek. Ezért az automatikus nyelvtan generálás napjainkban is a számítástudomány aktív kutatási területének számít.

A nyelvtanok automatikus generálása nem csak a természetes nyelvű szövegek feldolgozása során alkalmazható, számos olyan tudományterület létezik, ahol az egyes tevékenységek, elemek leírására új mesterséges nyelveket lehet bevezetni. Ilyen területek például a biológia, a kémia és a biokémia, ahol az egyes kölcsönhatásokat bonyolult, összetett mondatok formájában adják meg [1]. A nyelvtan ezen felül egy tömör leírását adja a nyelvnek, így akár tömörítési feladatokra is alkalmazható [2].

A szöveges adatok és dokumentumok feldolgozására, elemzésére számos szabvány és megközelítés létezik, de a formális nyelvek kezelésére ezek viszonylag kevés lehetőséget biztosítanak. A formális nyelvek azon belül a környezet független nyelvek, automatikus generálásával kapcsolatos kutatásaink során felmerült az igény a különböző módszerek közös környezetben történő implementálására, így lehetővé téve tesztelésüket, összehasonlításukat, elemzésüket. Jelen munkában a META keretrendszert [3] ismertetem, aminek a fő célja, hogy a különböző formális nyelvek feldolgozására használt módszereket keretbe foglalja.

A továbbiakban röviden ismertetem a formális nyelvek leírását, kiemelve a számunkra fontos környezet független nyelvek osztályát. Majd áttekintem a legismertebb jelenleg elérhető adatbányászati, szövegbányászati rendszerek fő funkcióit. A jelenlegi rendszerek vizsgálata után ismertetem a META rendszerrel szembeni elvárásokat és részletezem a fő tervezési lépéseket. Ezután a formális nyelvekkel kapcsolatos modult részletesen ismertetem, majd az irodalomból jól ismert algoritmusokat mutatok be. Végül összegezem a kitűzött és az elért eredményeinket, illetve ismertetem a rendszer továbbfejlesztési lehetőségeit.

*Phd hallgató - Miskolci Egyetem Általános Informatikai Tanszék

[1] Miskolcian Environment for Text Analysis

2. FORMÁLIS NYELVEK

Az egyes természetes és mesterséges nyelveket matematikai szempontból a formális nyelvek segítségével lehet modellezni, ahol egy adott véges ábécé ($\sum$) felett értelmezett nyelv ($L = \{\omega\}$) mint mondatainak ($\omega \in \sum^*$) halmazaként van megadva. Az egyes nyelveket a nyelvtanukkal (G) lehet megadni, és azt mondjuk, hogy az L_G nyelvet a G nyelvtan állítja elő. A nyelvtant a $\langle T, N, P, S \rangle$ négyes határozza meg, ahol:

- T a terminális szimbólumok véges halmaza
- N a nem terminális szimbólumok véges halmaza
- P a képzési szabályok halmaza ($P = \{\propto \rightarrow \beta \mid \alpha, \beta \in (T \cup N)^*\}$)
- S a mondat kezdő szimbólumok halmaza ($S \subseteq N$)

A formális nyelvek egyik jól ismert osztályozása a Chomsky hierarchia, amely az egyes nyelveket a képzési szabályaik alapján egyre szigorúbb csoportokba sorolja (lásd *1. táblázat*). A Chomsky hierarchia osztályai közül a természetes nyelvek leírására a CSG és a CFG nyelvek alkalmasak, ezek közül most csak a CFG nyelvekkel foglalkozok, mert ezekkel hatékonyan lehet műveleteket végrehajtani.

Nyelv	Megvalósítás	Képzési szabály
Rekurzívan felsorolható nyelvek	Turing gép	$\alpha \rightarrow \beta$
Környezet függő nyelvek (CSG)	Linear-bounded automata	$\alpha A \beta \rightarrow \alpha \gamma \beta$
Környezet független nyelvek (CFG)	Push-down automata	$A \rightarrow \alpha$
Reguláris nyelvek	Véges állapotú automata	$A \rightarrow aB$ vagy $A \rightarrow Ba$ és $A \rightarrow a$

1. táblázat Chomsky hierarchia

2.1. Sztochasztikus környezet független nyelvek

A környezet független nyelvek egy speciális csoportja a sztochasztikus környezet független nyelvek (Probabilistic Context Free Grammar) [4], ahol az egyes szabályoknál egy bal oldalhoz több különböző valószínűségű jobb oldal is tartozhat. A PCFG képzési szabályai a CFG nyelvekhez hasonlóan: $A \rightarrow \alpha$ alakúak, de minden szabály minden jobb oldalához meg lehet határozni a valószínűségét $Pr(A \rightarrow \alpha_i)$ úgy, hogy $\sum_{A \rightarrow \alpha_i \in P} Pr(A \rightarrow \alpha_i) = 1$. A PCFG előnye a CFG-vel szemben, hogy méri a bizonytalanságot is, illetve a PCFG nyelvtanok automatikus előállításához pozitív minták is elegendőek. Ezzel szemben a CFG nyelvtanok automatikus előállítása során negatív tanítómintákat is meg kell adni, aminek az előállítása időigényes feladat lehet.

3. ADATBÁNYÁSZATI ÉS NLP RENDSZEREK

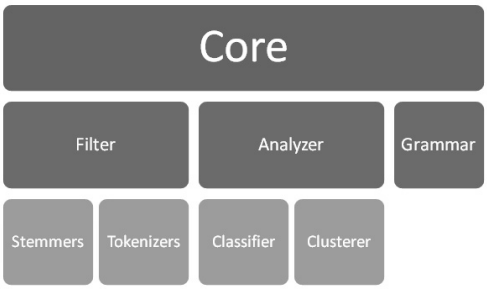
Az irodalomban számos módszer található a CFG és PCFG nyelvek generálására [5] és feldolgozására, illetve számos adatbányászati keretrendszer érhető el, melyek különböző mértékben szöveg feldolgozási funkciókat is megvalósítanak. Az egyik legismertebb ilyen keretrendszer a Weka ami egy adatbányászati és gépi tanulási keretrendszer. A Weka mellett számos adatbányászati alkalmazás és rendszer érhető el a piacon, de ezeket a terjedelemre való tekintettel nem ismertetem. A formális nyelvek feldolgozására egy minta alkalmazás a JavaCFG, amit oktatási céllal fejlesztettek a Columbia Egyetemen. Bár a megvizsgált keretrendszerek és alkalmazások számos szolgáltatást nyújtanak, amik segítenék a CFG generálás területén végzett munkát, de egyik rendszer sem elégíti ki teljes mértékben az igényeket.

A Weka az egyik legismertebb adatbányászati keretrendszer, ami számos osztályozó, klaszterező, előfeldolgozó algoritmust tartalmaz adatbányászati feladatok ellátására, viszont nem támogatja kellően a formális nyelvek modellezését. A rendszer a nyíltsága és jó dokumentációja miatt népszerű és elterjedt, ezért vettem alapul a META keretrendszer fejlesztése során. A Weka rendszerhez a könnyű használat érdekében egy egyszerű grafikus felhasználói felület is tartozik, ami segítségével könnyen lehet grafikusan szemléltetni az eredményeket.

Az irodalomban található számos mintaillesztő [6] algoritmus és nyelvtan generáló módszer [7,8] környezet független nyelvek számára. Az környezet független nyelvek generálásával kapcsolatban végzett legfrissebb eredményekről, tendenciákról jó áttekintés található D'Ulizia munkájában [5]. Ezeket a módszereket különböző programozási nyelveken implementálták és tesztelték. Az egyes módszerek jelentősen eltérhetnek egymástól mind alapötletükben, mind alkalmazott technikájukban. A cél egy közös környezetbe implementálni a legismertebb módszereket, hogy megfelelően tesztelni, összehasonlítani lehessen azokat.

4. META KERETRENDSZER

A META [3] egy formális nyelvi és szöveg feldolgozási keretrendszer, amely fő célja, hogy közös keretbe foglalja az egyes formális nyelveken végezhető módszereket. A rendszerrel szembeni alapvető elvárás, hogy lehetőséget nyújtson a formális nyelvek modellezésére, kezelésére és könnyen bővíthető legyen új módszerekkel. A rendszer ezen felül egyéb funkciókat is nyújt a felhasználók számára, mint például a naplózás. Mivel a nyelvtan generálás szorosan kapcsolódik a természetes nyelvek feldolgozásához és a szövegbányászathoz, ezért a keretrendszer lehetővé teszi, hogy később további szövegbányászati módszerekkel való kiegészítésre. De a rendszer jelenlegi verziója csak az egyes interfészeket definiálja, konkrét algoritmust nem valósít meg.



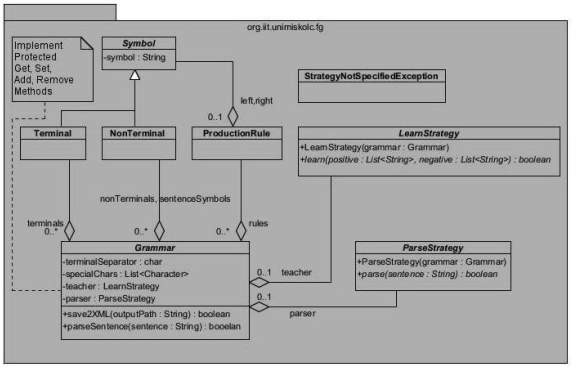
1. ábra A META keretrendszer struktúrája

Az ***1****. ábra* mutatja a META keretrendszer struktúráját, amin látható, hogy a keretrendszer a Core modul köré épül. A Core modul nyújtja a többi modul számára az alapvető szolgáltatásokat és adatstruktúrákat. Az egyes algoritmusokat a megfelelő modulok valósítják meg, például az előfeldolgozást végző különböző módszereket a Filter modul foglalja össze, amíg az elemző funkciókat az Analyzer valósítja meg. A nyelvtani módszereket a Grammar modul valósítja meg. Így a rendszerstruktúra lehetővé teszi számos módszer átlátható implementálását a rendszerbe.

4.1. A környezet független nyelvi modul

A Grammar modul a formális nyelvek leírását valósítja meg, valamint lehetőséget ad saját mintaillesztő és nyelvtan tanuló módszerek implementálására. A formális nyelvi csomag UML osztálydiagramja a ***2****. ábra* látható. A modul a különböző típusú nyelvtan elemeket külön osztályok segítségével valósítja meg. A Grammar osztály ezen osztályok kollekcióit tartalmazva valósítja meg a nyelvtan leírását. A bővíthetőség érdekében az egyes nyelvtanok különböző mintaillesztő és tanuló stratégiákat alkalmazhatnak, így az új módszer hozzáadása a megfelelő ősosztály leszármaztatásával érhető el. Az használni kívánt tanító illetve mintaillesztő módszert a Grammar objektumnak be kell állítani, a learn illetve a parse metódus első meghívása előtt. Ha a felhasználó elmulasztotta a stratégia meghatározását, akkor StrategyNotSpecified-Exception kivétel keletkezik a learn vagy parse metódusának meghívásakor.

A nyelvtanok ilyen fajta megvalósítása nagy szabadságot enged a saját módszerek implementációja során. A nyelvtanok modellezését segíti, hogy a képzési szabályok a rekurzívan felsorolható nyelvek szabályait valósítják meg, azaz a szabályok $\alpha \rightarrow \beta$ alakúak. A programozónak kell felügyelnie, hogy milyen típusú nyelvtant állít elő. A szabályok között a rendszer megkülönbözteti az egyszerű és a sztochasztikus képzési szabályokat, valamint megkülönbözteti az egyszerű és a sztochasztikus nyelvtanokat, mivel a szabályaik kezelése jelentősen eltér egymástól.



2. ábra A formális nyelvi modul struktúrája

A Grammar csomag egy részcsomagja a CFG csomag, ami a környezet független nyelvek feldolgozásához kapcsolódó módszereket foglalja össze. A csomag számos az irodalomban megtalálható módszert megvalósít, többek között mintaillesztő algoritmusok közül a CYK algoritmust és különböző tanító algoritmusokat, mint például a TBL [8], az ITBL [7], és az InduktívCYK [9] algoritmust. Az egyes módszerek implementálásával láthatóvá vált, hogy a META keretrendszer képes összefogni a különböző módszereket, így teljesíti a legfőbb elvárásokat.

5. TOVÁBBFELJELSZTÉSI LEHETŐSÉGEK

A META keretrendszer jelenlegi verziója az egyes módszereknek API-n keresztül történő elérését támogatja. Bár a jelenlegi rendszer megfelelő szolgáltatások biztosít különböző módszerek implementálására és a

rendszer kiterjesztésére, az egyes módszerek tesztelése, a futási információk megjelenítése, elemzése kényelmetlen és költséges feladat. A könnyebb használat érdekében egy korszerű, áttekinthető GUI-s alkalmazás készítését tervezem.

A grafikus alkalmazás fejlesztése mellett további cél, az irodalomban fellelhető [5] algoritmusok implementálása a META keretrendszerbe. A későbbiekben szeretném részletesen elemezni a különböző módszerek hatékonyságát, időköltségét.

A különböző módszerek elemzésén túlmutatóan saját CFG, PCFG generáló módszer kidolgozását tűztem ki célul, melyet a META keretrendszerbe fogok implementálni. A tervezett módszert így számos az irodalomban korábban megjelent algoritmussal össze tudom majd hasonlítani.

6. ÖSSZEFOGLALÁS

A tanulmányban egy új, korszerű formális nyelvi keretrendszert és annak környezet független nyelvi modulját mutattam be. A formális nyelvek matematikai formalizmusának rövid áttekintését követően ismertettem néhány elérhető, hasonló rendszert. Az egyes rendszerek előnyeit, lehetőségeit tömören ismertettem, kitértem az egyes rendszerek korlátaira. Megállapítottam, hogy egyik ismertetett rendszer sem nyújt teljes körű támogatást a formális nyelvek kezelésére. Az ismertetett rendszerek áttekintése után megfogalmaztam a META rendszerrel szemben támasztott követelményeket, igényeket.

Az igények összegzése után bemutattam a tervezett rendszer architektúráját, ismertettem a felépítés előnyeit, az egyes fő modulokat. Külön kiemeltem a rendszer lehetőségeit és röviden kitértem a szövegbányászati modulra. A rendszer áttekintése után a formális nyelvi Grammar modult részleteztem. A Grammar modul ismertetése után a környezet független nyelvtanok kezelésére szolgáló algoritmusok implementációját ismertettem. A CFG modul számos az irodalomban fellelhető, jól ismert módszert tartalmaz. A rendszer bemutatását követően a továbbfejlesztési lehetőségeket, valamint a további felhasználási lehetőségeket foglaltam össze.

7. KÖSZÖNETNYÍLVÁNÍTÁS

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

8. IRODALOMJEGYZÉK

[1] **Gavrilis, D. and Tsoulos, I. and Dermatas, E.** Evolutionary Grammar Induction for Protein Relation Extraction.

[2] *Compression by induction of hierarchical grammars.* **Nevill-Manning, C.G. and Witten, I.H. and Maulsby, D.L.** Data Compression Conference : IEEE, 1994.

[3] *META: A novel grammar induction and text mining framework.* **Tóth Zs., Kovács L.** XXVI. MicroCAD konferencia - Miskolc : Miskolci Egyetem, 2012.

[4] *PCFG models of linguistic tree representations.* **Johnson, M.** 4, Computational Linguistics : MIT Press, 1998., 24. kötet.

[5] *A survey of grammatical inference methods for natural language learning.* **Arianna D'Ulizia, Fernando Ferri, Patrizia Grifoni.** 1-27, Artificial Intelligence Review : Springer, 2011., 36. kötet.

[6] *A generalized CYK algorithm for parsing stochastic CFG.* **Chappelier, J.C. and Rajman, M.** First Workshop on Tabulation in Parsing and Deduction : Citeseer, 1998.

[7] *Improved TBL algorithm for learning context-free grammar.* **Jaworski, M. and Unold, O.** Proceedings of the International Multiconference on ISSN : ismeretlen szerző, 2007.

[8] *Ga-based learning of context-free grammars using tabular representations.* **Sakakibara, Y. Kondo, M.** Machine Learning - International Workshop then Conference : Morgan Kaufmann Publishers Inc., 1999.

[9] *Synthesizing context free grammars from sample strings based on inductive CYK algorithm.* **Nakamura, K. Ishiwata, T.** Grammatical Inference: Algorithms and Applications : Springer, 2000.