

A szövegelemzés gépi módszere

(bevezető)



Problémafelvetés: alkalmazási igények

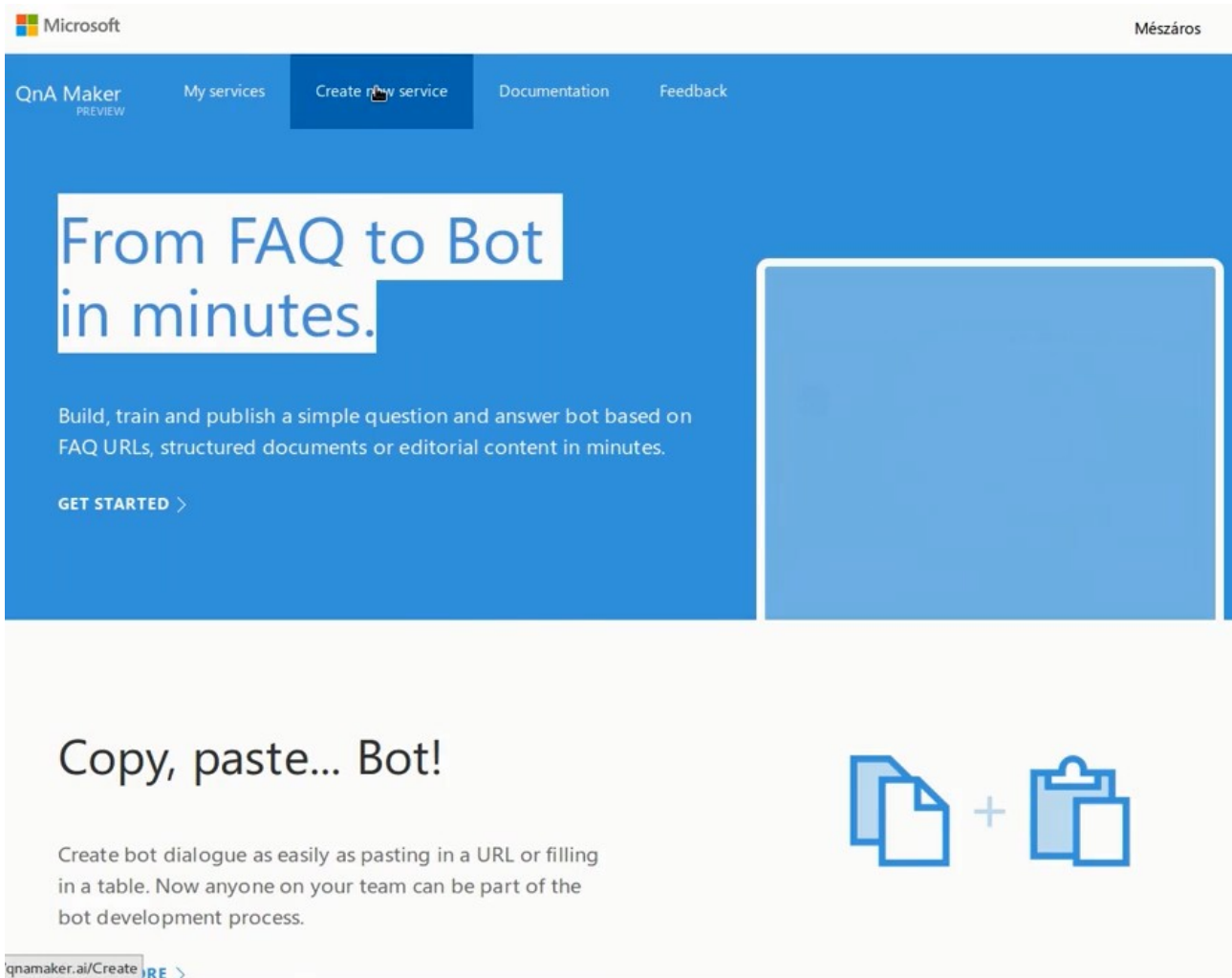
- „Szeretnék megtudni valamit. Segít ebben a számítógép?”
 - internetes keresés
 - digitális könyvtárak
- „Beszélni akarok a gépekkel, nem nyomogatni azokat”
 - Apple Siri, Google Voice, Microsoft Cortana, Amazon Alexa stb.
- „Nem értem a program kezelését. Elmondhatom, mit szeretnék?”
 - komplex rendszerek → összetett felhasználói felületek
 - tudásbevitel, tudáskinyerés
- „Szeretném, ha a gép lefordítaná nekem ezt a szöveget!”
 - mit közölnek mások
 - hogyan mondhatom el másik nyelven

A természetes nyelvű kommunikáció

- A természetes nyelv az emberi kommunikáció meghatározó formája.
 - mindenki ismeri és használja
 - tudásleírás és -megosztás
 - a segítségével befolyásolhatjuk a körülöttünk levő világot *
- Miért nem kommunikálnak velünk a gépek természetes nyelven?
 - Mennyivel egyszerűbb lenne
 - bonyolult felhasználói felületek,
 - mesterségesen létrehozott (programozási) nyelvek,
 - gyenge kifejezőerejű vagy nem létező kommunikációs interfészek helyett.
 - Látunk erre példákat, de valahogy nem mindennaposak és nem átütőek.

Miért?

Microsoft Q&A Bot Maker



The screenshot shows the Microsoft QnA Maker web application. At the top, there's a navigation bar with the Microsoft logo, the user name 'Mészáros', and links for 'QnA Maker PREVIEW', 'My services', 'Create new service' (highlighted), 'Documentation', and 'Feedback'. The main content area has a blue background with the text 'From FAQ to Bot in minutes.' and a subtext 'Build, train and publish a simple question and answer bot based on FAQ URLs, structured documents or editorial content in minutes.' Below this is a 'GET STARTED >' button. To the right is a large, empty light blue box. The bottom section has a white background with the heading 'Copy, paste... Bot!' and a subtext 'Create bot dialogue as easily as pasting in a URL or filling in a table. Now anyone on your team can be part of the bot development process.' To the right of this text is an icon showing two document pages with a plus sign between them. At the bottom left, there's a link 'qnamaker.ai/Create' followed by a blue 'RE >' button.

Problémafelvetés a megvalósítás szemszögéből

- Mitől olyan nehéz a szövegekkel bánni?
- Mit jelent számunkra egy természetes nyelvű közlés (szöveg)?
 - dekódolás
 - értelmezés
- Mire képesek mindebből a gépeink?
 - dekódolni
 - ... ? ...
- Értelmezés?
 - szavak jelentése + a közlés kontextusa
 - a jelentés befogadása, beépítése a tudástárunkba
- Ha ennyire nehéz a helyzet, mi értelme az egésznek?

A nehézségek részletesen

- Beszédfelismerés (audio jelek feldolgozása, szöveggé alakítása)
 - sokféle hang, stílus, beszédhibák - néha még nekünk sem triviális
- A nyelv szintaktikai szabályainak ismerete
 - nem rögzített, térben és időben is változik
- Szókincs
 - nem ismerjük a teljes szókincset, térben és időben változik
- Értelmezés
 - szavak értelme, szavakból épített kifejezések értelmezése
 - a mondatok értelmezése, beleértve az írásjelek módosító hatását is
 - a közlés kontextusának hatása az értelmezésre
- Párbeszédkezelése
 - egy közlés értelme alapvetően függhet a korábbiaktól („Nem.”)
- Hivatkozások feloldása
 - „ő volt az, aki...”, „nem, én”
- A válasz előállítása (mit és hogyan mondunk el)
 - karakterek összefűzése értelmes mondattá
- Az előállított válasz hangjelekké alakítása (kiejtés, hangsúly)

NLP

- Természetes nyelvű szövegek feldolgozása
Natural Language Processing (NLP)
- Az MI egyik legrégebbi kutatási területe.
 - beszédfelismerés, szintaktikai elemzés, szemantikai értelmezés, párbeszéd-kezelés, mondatgenerálás, beszédszintézis, gépi fordítás, helyesírás-ellenőrzés, kérdésmegválaszoló rendszerek stb.
 - számos **könyv** és **publikáció**
- State of the art

Az NLP eredményeit sokfelé látjuk az iparban és mindennapjainkban.

 - számos helyen találkozunk az alkalmazásaival
 - természetes nyelvű párbeszédre képes rendszerek
(ügyfélszolgálatok, tanító rendszerek, segítő és szórakoztató robotok stb.)
 - keresőrendszerek
 - szó- és mondatelemző szoftverek (pl. helyesírás)
 - szövegelemző és tudáskinyerő rendszerek (pl. termékértékelések, marketing stb.)
 - gépi fordítás (pl. Google Translate)

Példák projektjeinkből

Hogyan működnek ezek a rendszerek (madártávlatból)?

A természetes nyelvű szövegek gépi feldolgozása

- Mi a természetes nyelv
 - mi a célja?
 - mire használjuk?
 - hogyan sajátítjuk el?
- Mire használható a számítógép?
 - „ügyes” írógép
 - „buta” tároló
 - keresőgép (lokális, internetes)
 - szemantikus szolgáltatások
 - természetes nyelvű interfészek
 - fordítógép
 - ...

Hogyan kezdünk neki?

- Szövegfeldolgozás

- szabad / strukturált szöveg?
- célja?

- Szövegkeresés

- Szövegfüzérek egy halmazára illeszkedő kifejezés
- egy *minta* előfordulása(i)t keresi egy *korpuszban*
- pl. a /villany/ → „villany”, „villanykapcsoló”
- Alapvető minták

\b break \d digit \D not digit [abc] set

\s whitespace \w word char \W not word [^abc] neg. set

{min_num, max_num} számosság, pl. \S{3} \w{1,3}

tokenek csoportosítása (abc) Hivatkozás \1 \$1

tesztek (.=abc) (?!abc) (villanyt|lámpát) (?.*sötét.*)

- Sokféle eszköz: grep, awk, sed, Java, C#, PHP, Python stb.

Webes [Tesztelőeszköz](#) és [gyakorló példák](#)

Az információelérés támogatása

- Mennyire hatékony a kereső?

- Milyen gyors a regex keresés?

Mi a helyzet, ha mondjuk 11 millió szó között keresünk?

- Mennyire pontos a keresés?

- visszaadás (recall)

Konstantinápolyal hány írás foglalkozik? Mennyit ad vissza a kereső?

- pontosság (precision)

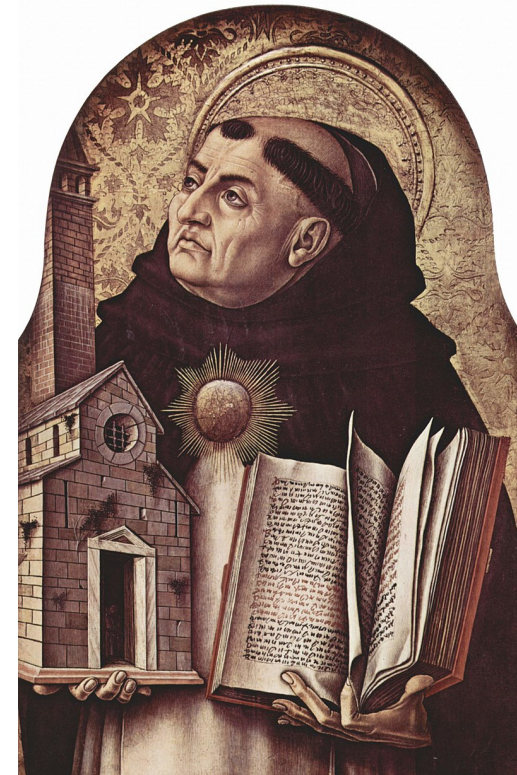
A „néném” szóra keresve visszakapok nem releváns találatokat is?

Mit tehetünk a helyzet javítása érdekében?

Roberto Busa – *Index Thomisticus*



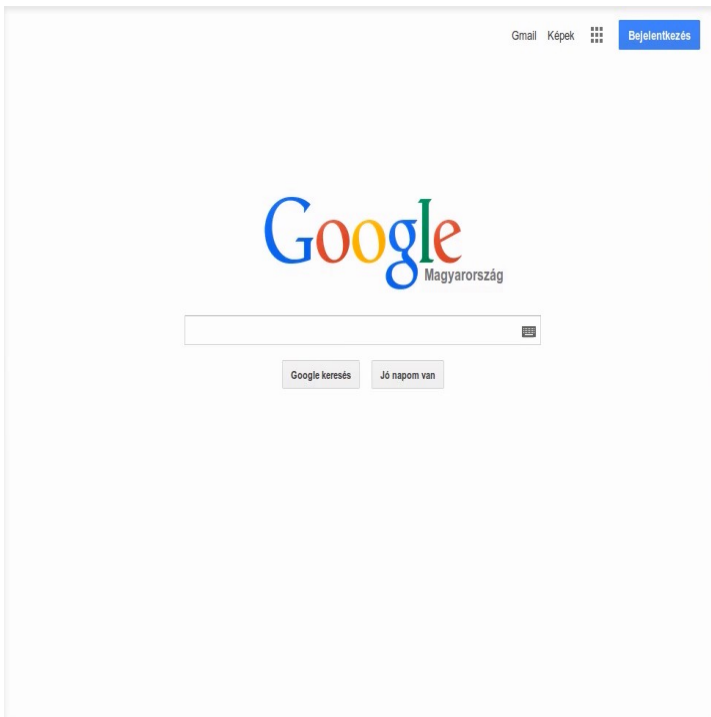
Distant reading – távtartó olvasás



Aquinói Szent Tamás

Hogyan kezdünk neki?

- Kitől tanulunk?
 - distant reading (távartó olvasás)
 - katalogizálás és konkordancia listák
 - statisztikai módszerek

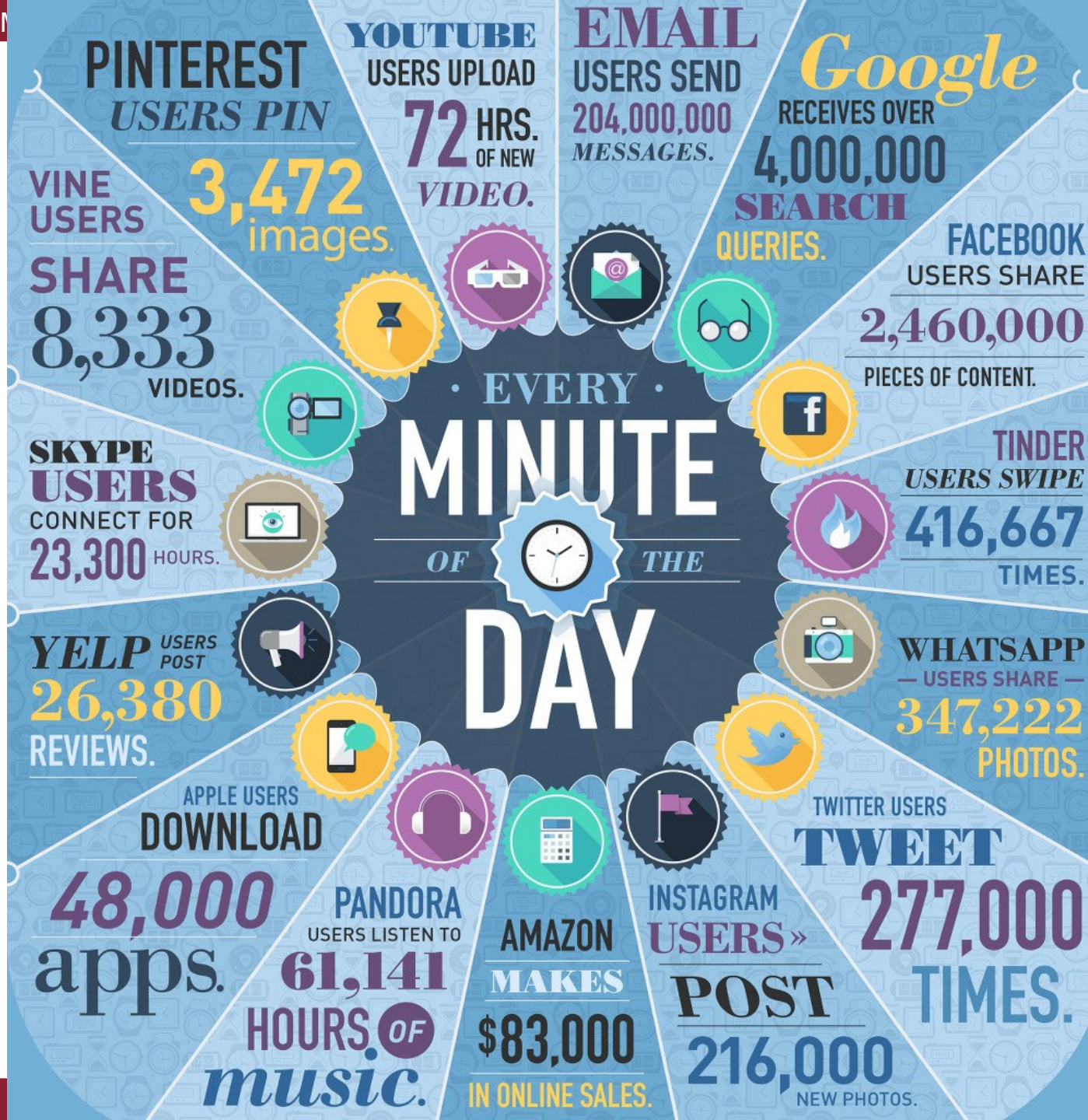


Az információelérés problémái

- Meg tudjuk fogalmazni, mire van szükségünk?
 - Elegendő a kulcsszavak felsorolása?
- Megtaláljuk, amit szeretnénk? *(recall)*
 - Vajon a Google a web mekkora részét „látja”?
 - Lépést tartunk a tárolt információk mennyiségi növekedésével?
- Azt kapjuk, amit szeretnénk? *(precision)*
 - 16 000 000 találat között hány nem releváns tétel van?

Mi a problémák forrása?

Sok
az
adat



Forrás: <https://www.domo.com/blog/2014/04/data-never-sleeps-2-0/>

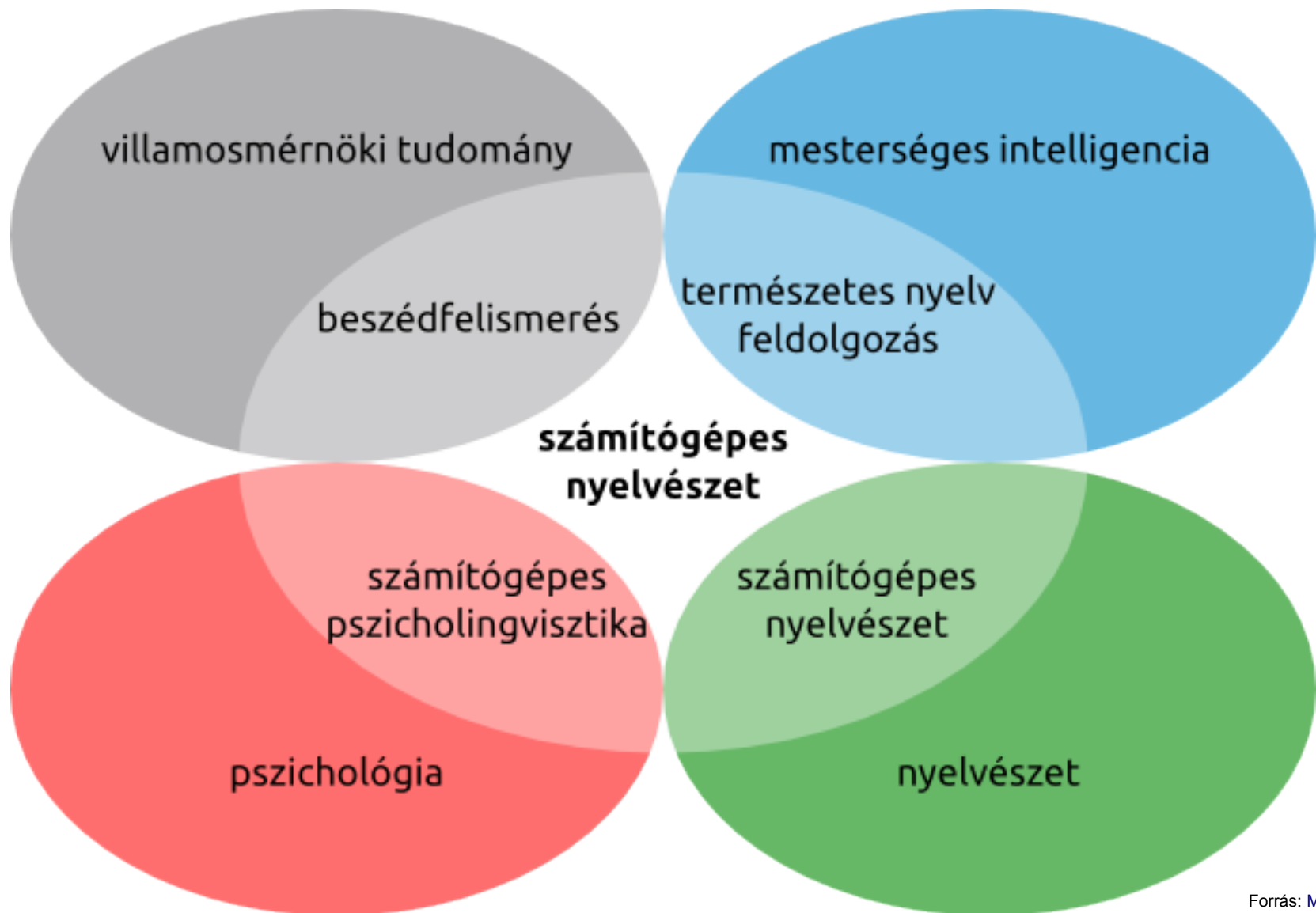
A számítógép nem érti a tartalmat

- Természetes nyelvű közlések
- Komoly háttértudás szükséges a megértésükhöz
- Szemantikus tárolás – a szöveg korlátozott „megértése”
 - megjelöljük (XML)
 - értelmezzük (ontológia)
 - kikövetkeztetjük (logika)
- A szemantikus web koncepció (90-es évek vége !!)
 - W3C
 - strukturált dokumentumok
 - + szemantikus jelölés
 - + fogalmi rendszerek
 - + logika

Hol a szemantikus web?

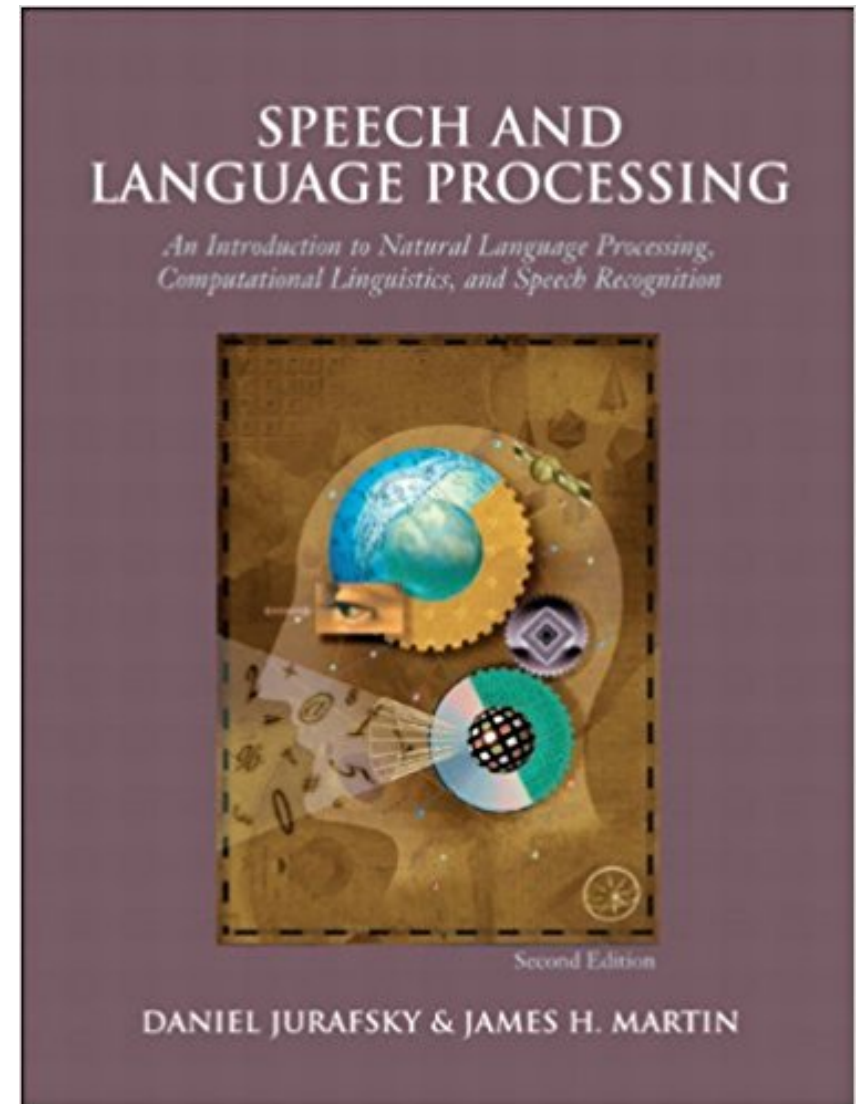
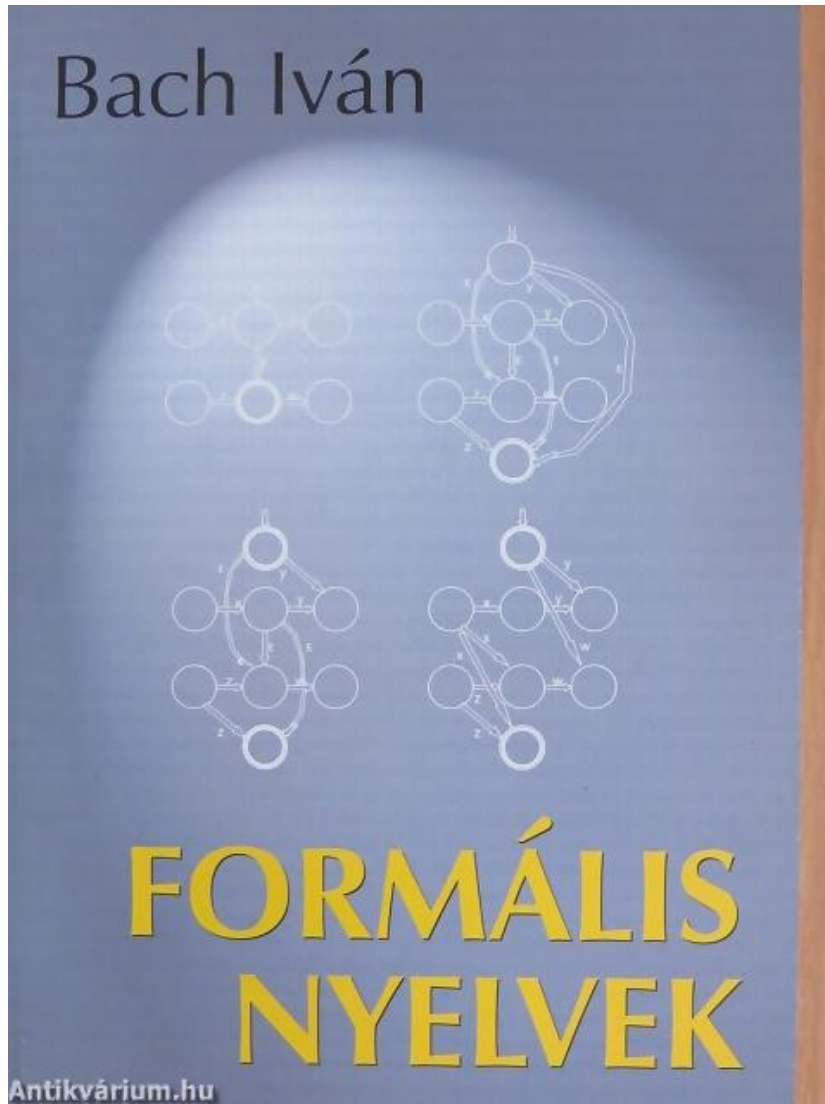
- Eltelt 15-20 év...
 - sok szabvány
 - mindennapos technológia (XML)
 - elérhető, elterjedt
- A web tartalmait szinte ugyanúgy HTML-ben tároljuk
- Időközben...
 - újabb és újabb tartalomkezelő rendszerek
 - dinamikus webrendszerek (Javascript, Ajax stb.)
 - elképzelhetetlen mértékben nő a tárolt adatmennyiség
 - nem lett egyszerűbb az információk megtalálása

Számítógépes nyelvészet



Forrás: MI Almanach

Számítógépes nyelvészet



Fogalmi bevezető

- Alfabéta (karakterkészlet) Σ
 - jelek (építőelemek) véges halmaza

- Füzér (karaktersorozat) és mondat (S) α, β ε – üres füzér
 - az alfabétából generálható jelsorozat
 - „véges, de nem korlátos hosszúságú” (mindig van vége, de akármilyen hosszú)
 - nyelv = összes lehetséges mondat (ha véges)
 - dokumentum: mondatok halmaza, **korpusz**: dokumentumok (nagy) halmaza

- Nyelvtan (grammatika) $G = (N, \Sigma, P, S)$
 - nem véges nyelvek leírására (levezetésére, generálására)
 - N : grammatikai szimbólumok (nemterminális) A, B, C
 - Σ : a karakterkészlet szimbólumai (terminális) a, b, c
 - A teljes szimbólumkészlet ($N \cup \Sigma$) α, β ($N \cap \Sigma = \emptyset$)
 - S : mondatszimbólum $S \in N$
 - P : levezetési (produkció, újraíró) szabályok $\alpha \rightarrow \beta$
 szimbólumsorozat cseréje, ahol α legalább egy nemterminálist tartalmaz
 - Levezetés: szabályok alkalmazása a grammatikai szimbólumok elfogyásáig
 Az S -ből kiindulva eldöntjük, hogy egy szimbólumsorozat eleme-e a nyelvnek.

A gyakorlatban...

- Az alfabéta reprezentációja
 - ASCII (az első, de nem tűnt el), pl. ISO-8859-2 (1 byte)
 - Unicode (új, elterjedt), pl. UTF-8 (1, 2 vagy 4 byte)
- Karaktorsorozat (füzér)
 - sokféle adattípus: `char *`, `char[]`, `string`, `Text` stb.
 - sokféle kezelőfüggvény: hossz, összefűzés, darabolás, keresés stb.
`echo strlen("ő")` vs. `mb_strlen("ő");`
- Dokumentum
 - de facto és de jure szabványos formátumok (DOCX, HTML, XML stb.)
 - szövegkezelő programok széles skálája (awk, sed, grep, perl stb.)
- Dokumentumtár, korpusztár
 - fájlrendszer
 - adatbázisok (pl. noSQL)
 - tartalomkezelő-rendszerek

Nyelv és nyelvtan

- Egy nyelv megadása
 - alfabéta
 - az alfabétából felépíthető érvényes füzérek (akár végtelen) halmaza
 - generálási szabályok a füzérek felépítéséhez (és érvényességük ellenőrzéséhez)
 - nyelvtan: a nyelvet meghatározó szabályok véges halmaza
- Szabályok
 - **kifejezésstruktúra**: a füzérek részfüzérekre (kifejezésekre) bonthatók
 - „kapcsold fel a piros lámpát” → „kapcsold”, „fel”, „a”, „piros”, „lámpát”
 - „kapcsold fel a piros lámpát” → „kapcsold fel”, „a piros lámpát”
 - a kifejezésekhez grammatikai szimbólumokat rendelünk
 - UTASÍTÁS → „kapcsold fel” NÉVELŐ → „a” JELZŐ → „piros” TÁRGY → „lámpát”
 - a grammatikai szimbólumokból újabb grammatikai szimbólumokat építünk
 - MILYEN_TÁRGY → NÉVELŐ JELZŐ TÁRGY
 - UTASÍTÁS → PARANCs MOD PARANCs → „kapcsold” MOD → „fel” | „le”
 - ezek az ún. **átíró szabályok**
 - egy teljes mondat grammatikai szimbóluma az S
 - **levezetési** avagy **szintaxisfa**
keressünk azon szabályokat, amelyekkel egy adott füzér S-re alakítható

Egy példa nyelvtan

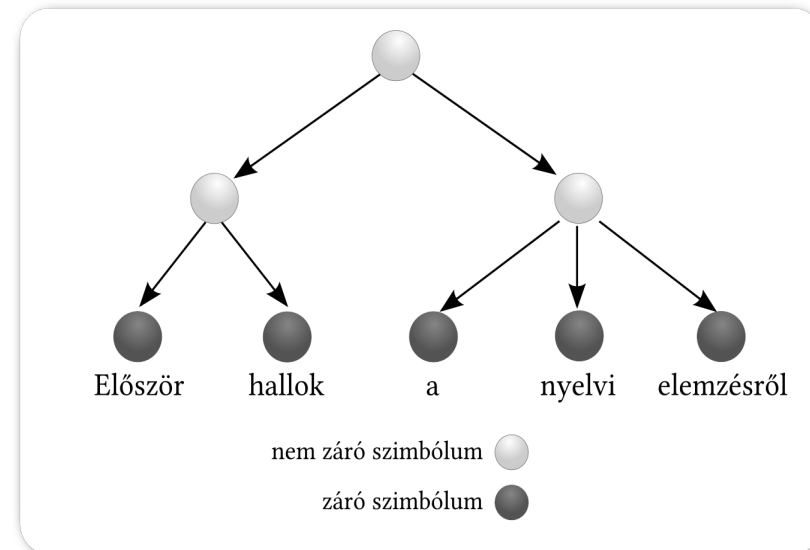
'S' => 'QW NP1 REL NP2',
 'QW' => 'melyik',
 'NP1' => 'főnév',
 'REL' => 'rendelkezik | nem rendelkezik',
 'NP2' => 'vonzattal | VJ vonzattal',
 'VJ' => 'MJ | VT | MJ VT',
 'VT' => 'birtokos | ban,ben |ról,ről | hoz,hez,höz
 |ból,ből |vhonnan |vhova |vmikor |val,vel |
 tól,től |ért,miatt |nak,nek |ra,re |vmilyen |
 vmennyi |n |után |között |ellen |szemben |
 mellett |belül |felett |íránt |be |felé',
 'MJ' => 'MJ1 | MJ2 | MJ1 MJ2',
 'MJ1' => 'fakultatív | kötelező',
 'MJ2' => 'élőre vonatkozó | élettelenre vonatkozó'

Szintaktikai elemzés

- Adott bemenethez keresünk egy levezetést
Egy bemeneti fűzérhez keressük átírószabályok azon sorozatát, amellyel eljutunk a terminálisokig.

`Parse(bemenet, G, S)`

- Ez olyan fát ad vissza, amelynek
 - gyökerében az S nyelvtani szimbólum áll
 - leveleiben az adott bemenet terminális szimbólumai vannak
 - belső csomópontjai nyelvtani (G) szimbólumok
 - egy csomópont gyerekei a benne található szimbólumot előállító valamely átíró szabály szimbólumsorozatának elemei balról jobbra



Elemzési algoritmusok

- Hogyan állítsuk elő az elemzési fát?
 - keresési feladat
 - fentről (S) lefelé vagy lentől felfelé
 - állapotátmeneti függvény: új csomópontok előállítása
 - célteszt: elkészült-e a teljes fa
- Problémák (ízelítő)
 - bal-rekurzív szabályok
 - $X \rightarrow X \dots$ (fentről lefelé elemezve végtelen ciklust okozhat)
 - komplexitás
 - keresés közben mennyit kell visszalépni a keresési térben
 - milyen információ vész el eközben
 - és kezelése
 - mennyire tekinthetünk előre a keresés során
 - hogyan őrizhetjük meg az átmeneti részeredményeket visszalépéskor
 - dinamikus programozás $\rightarrow$ hatékonyabb elemzők
 - rekurzív algoritmusokra jó
 - részproblémákat old meg, eredményeiket tárolja
 - pl. Early, CKY, **diagrammelemző** (chart parser)

Szemantikai értelmezés

A szintaktikai elemzés után szeretnénk megtudni a közlés jelentését.

- A füzér jelentéstartalmának előállítása
 - valamilyen tudásreprezentációs nyelven (pl. Prolog, LISP, RDF stb.)
 - API-hívások formájában (lásd majd Alexa)

Egyértelmű?

- A többértelműség feloldása (disambiguation)
 - a legtöbb kifejezésnek több értelme van
 - a közlő és a befogadó közös háttértudása segít a feloldásban
 - erősen valószínűségi alapú a feloldás
 - az elemzés többféle értelmezést állít elő
 - valamilyen bizonyossági mérték szerint döntünk

Szintaktikai elemző készítése

- Nem akarunk írni → generáljunk

- ANTLR

- elemzőgenerátor (rekurzív fentről-lefelé, dinamikus)

nyelvtan → **ANTLR** → elemzőprogram forráskód (sokféle nyelvre)

input → **elemzőprogram** → elemzési fa

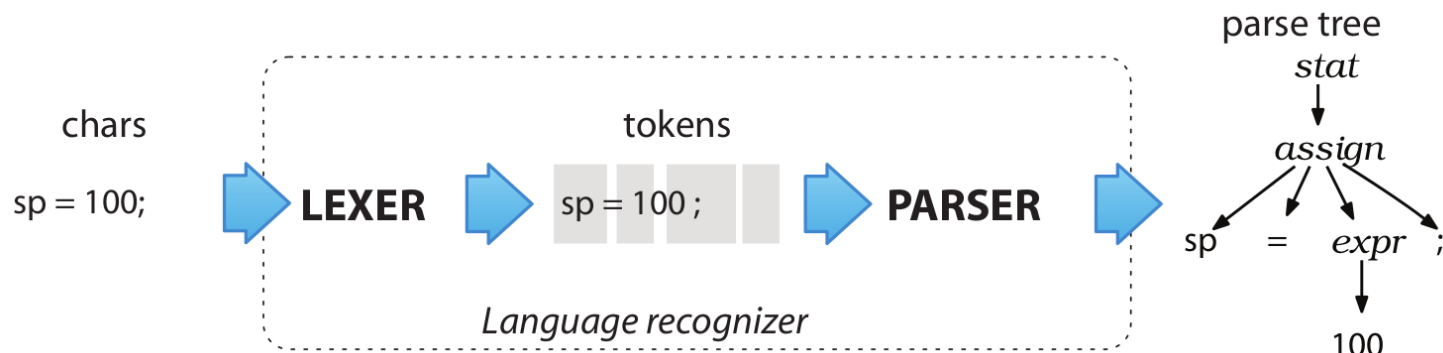
- a nyelvtan egy metanyelven írható le

stat : assign ;

assign : ID '=' expr ';' ;

ID : [a-z]+ ; (reguláris kifejezés)

expr : expr '*' expr | expr '+' expr | INT ; (rekurzív)



Forrás: [The Definitive ANTLR 4 Reference](#)

ANTLR kódgenerálás

ArrayInit.g4

```
grammar ArrayInit;
init : '{' value (',' value)* '}' ;
value : init
      | INT
      ;
INT : [0-9]+ ;
WS : [ \t\n]+ -> skip ;
```



ArrayInitParser.java

ArrayInitLexer.java

ArrayInit.tokens

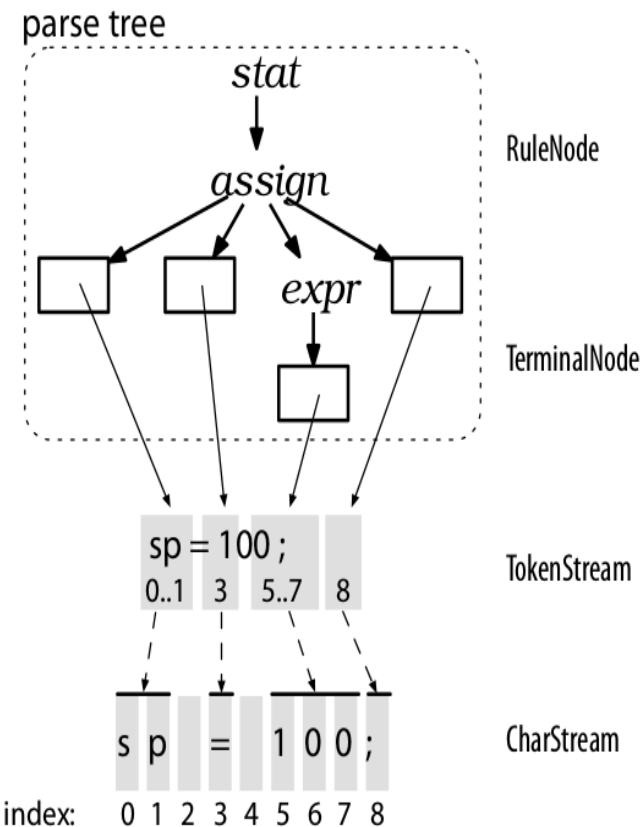
ArrayInitLexer.tokens

ArrayInitListener.java

ArrayInitBaseListener.java

Mit kezdünk az elemzési fával?

- Adatstruktúrák

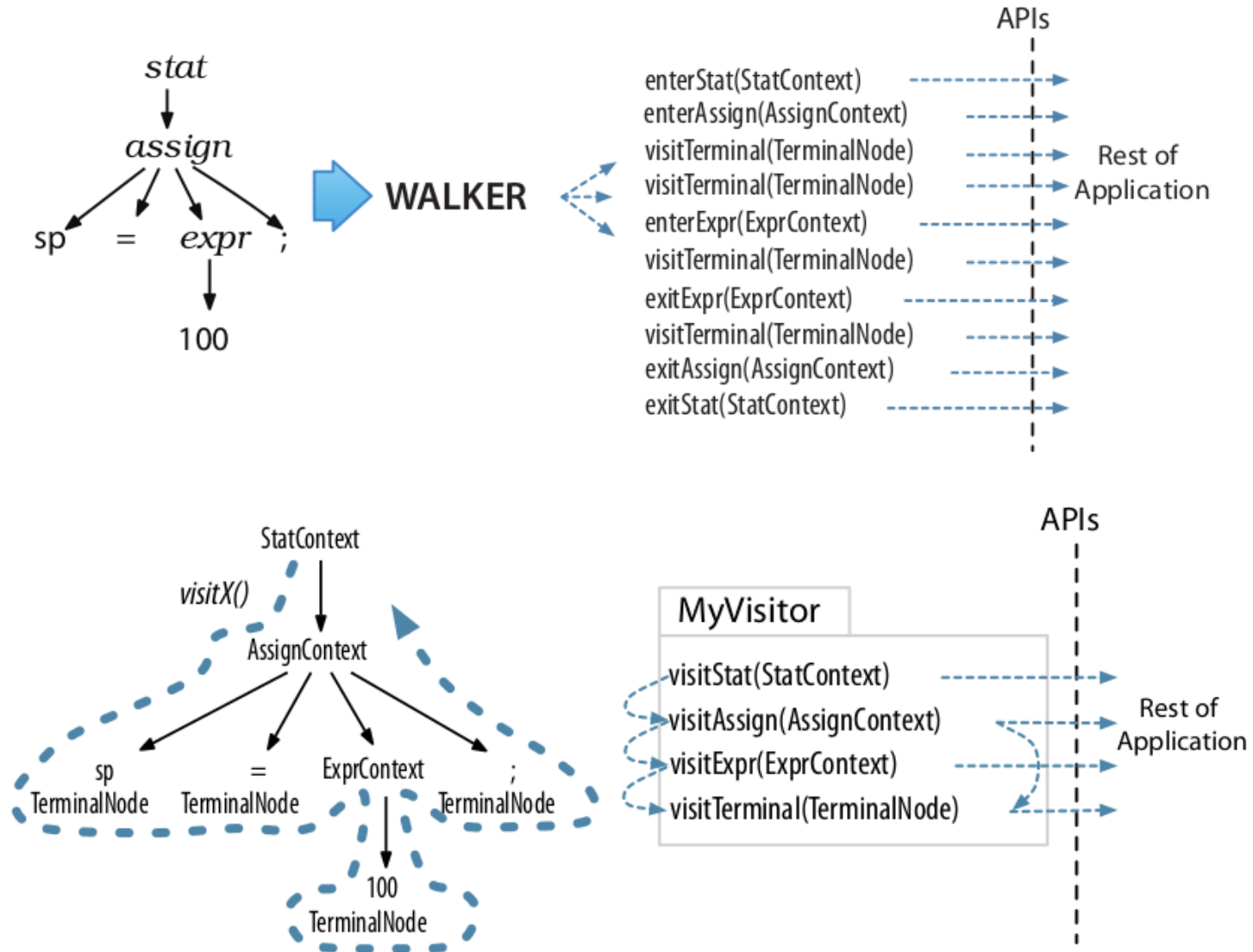


Forrás: The Definitive ANTLR 4 Reference

- Eljárások

- Listener interfész
fabejárás közben eseményeket generál
`enterAssign(AssignContext)`
ezekhez eseménykezelőket készíthetünk
~ SAX XML elemző
- beépített fabejáró (`ParseTreeWalker`)
eseményeket generál
- Visitor interfész
a fa bejárását meghatározó metódusok
`visitAssign(AssignContext)`
a bejárást mi programozzuk

Listener vs. Visitor



Forrás: The Definitive ANTLR 4 Reference

Milyen eszközöket próbálhatok ki?

Az ANTLR kipróbálása

- Eszközök: VS code + antlr.code-workspace + antlr4 + Java + grun

- Készítsen egy egyszerű nyelvtant világításvezérlésre!

Kapcsold fel / le lámpát / világítást (hol?)

- Generálja és fordítsa le az elemzőprogramot!

```
antlr4 iHomeAgent.g4 && javac iHomeAgent*.java
```

- Tesztelje a nyelvtant!

```
grun iHomeAgent start -tree
```

```
grun iHomeAgent start -gui
```

- Készítsen egy egyszerű Listener programot!

```
java iHomeAgentDemo teszt.txt
```

Eszközök: NLTK

- Természetes nyelvű szövegek feldolgozása és elemzése Python nyelven
 - korpuszok
 - szövegelemző könyvtárak (statisztikai, NLP)
 - széles körben elérhető
 - nyílt forráskódú
 - Könyv: [Natural Language Processing with Python](#)
- Elérhető eszközök
 - tokenizálás
 - szintaktikai elemzés
 - POS-jelölés
 - szótövesítés
 - ...

[Bevezető az eszközök használatába](#)

Próbáljuk ki az NLTK-t!

- A korpusz betöltése

```
files = "text-mining-corpus\/*.txt"
corpus = PlaintextCorpusReader(".", files, encoding='utf-8')
```

- Konkordancia

```
text = Text(corpus.words())
text.concordance("néném")
text.similar("néném")
```

- Gyakoriság, együttállás

```
fdist = FreqDist(text)
fdist.most_common(20)
text.collocations()
```

- Stopszavak

```
text2 = [word for word in text if word not in
stopwords.words('hungarian')]
```

- Szótővesítés

```
stemmer = SnowballStemmer('hungarian')
text3 = [stemmer.stem(word) for word in text]
```

Próbáljuk ki: véleménybányászat

- Sentiment Analysis
 - mit gondolnak a vásárlók egy termékről?
- DIY: kézi skálabeállítás (szabályalapú)
 - szavakhoz érzelmi tölteteket rendelünk
 - pl. negatív: rossz, vacak, kukába való ← szabályok
 - pozitív jelzők: kiváló, remek, jó stb. ← szabályok
 - határozzuk meg a fenti szavak előfordulását a szövegben
- Második nekifutás: a skála tanulása
 - tanítóminta-halmaz összeállítása
 - "szöveg 1": "pozitív", "szöveg 2": "pozitív", "szöveg 3": "negatív" ...
 - neurális háló határozza meg az egyes kategóriák legjellemzőbb szavait
 - Vannak előre betanított osztályozók is
- Eszközök
 - R Studio: Syuzhet, RStentiment
 - NLTK: Vader

MI a felhőben (SaaS, PaaS)

- Tudáskinyerés (gépi tanulás) MLaaS

Google „deep learning” ML Engine

Microsoft Azure ML Studio

Amazon Machine Learning

IBM Watson Analytics

- Kép- és videófeldolgozás

Google Vision API Amazon Rekognition

Clarifai

Watson Visual Recognition

Azure Computer Vision API

Microsoft Emotion API

- Szövegértés, párbeszédkezelés, gépi fordítás

Microsoft LUIS

Google API.ai

Facebook Wit.ai

Amazon Alexa

Amazon Transcribe

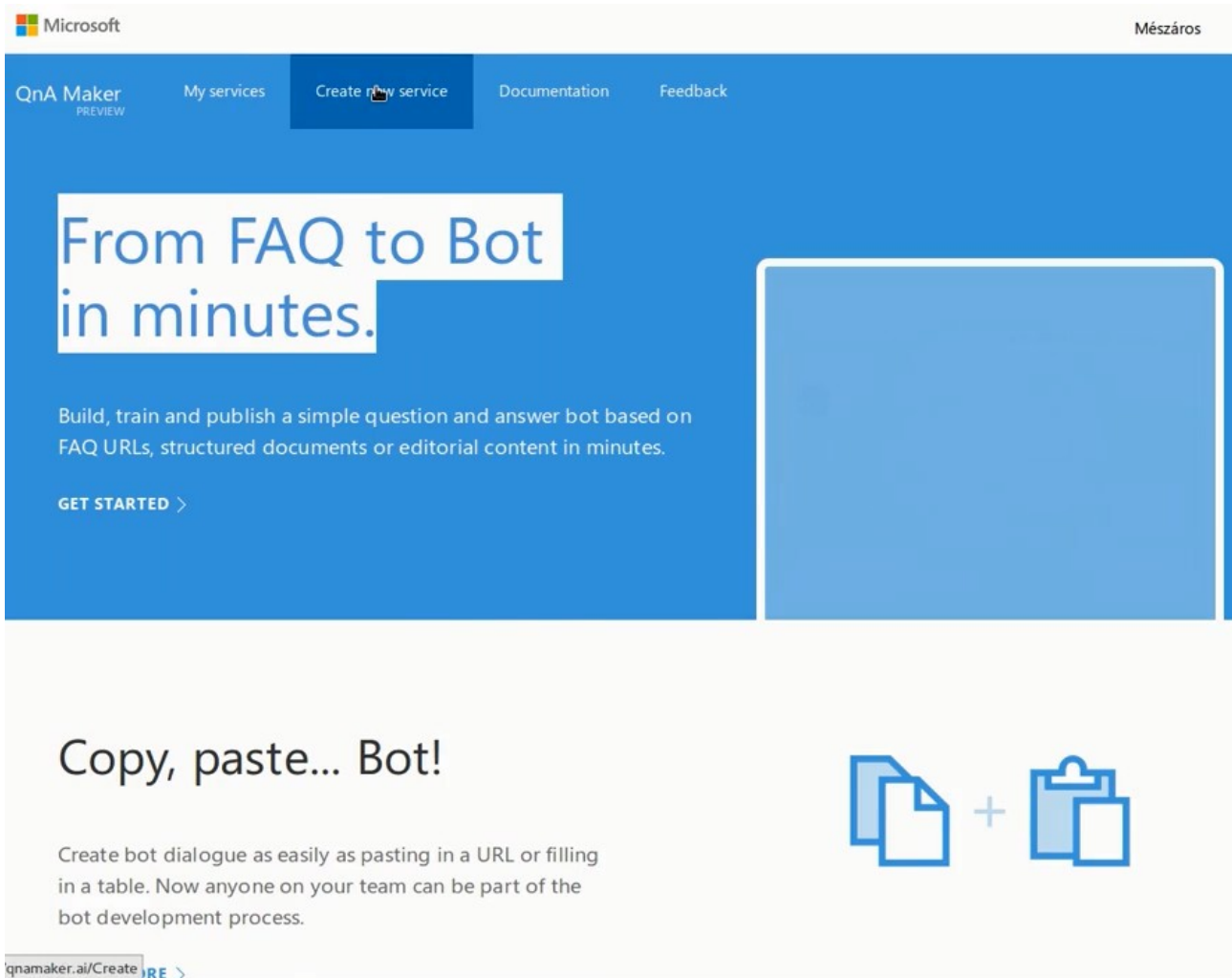
Amazon Comprehend

Amazon Translate

Google Translate

IBM Watson Conversation

Microsoft Q&A Bot Maker



The screenshot shows the Microsoft QnA Maker web application interface. At the top, there is a navigation bar with the Microsoft logo on the left and the user name 'Mészáros' on the right. Below the navigation bar, there are several tabs: 'QnA Maker PREVIEW', 'My services', 'Create new service' (which is highlighted with a dark blue background and a small icon), 'Documentation', and 'Feedback'. The main content area has a blue background. On the left, there is a white box containing the text 'From FAQ to Bot in minutes.' Below this, it says 'Build, train and publish a simple question and answer bot based on FAQ URLs, structured documents or editorial content in minutes.' and a 'GET STARTED >' button. On the right, there is a large, empty white box with a blue border. Below the main content area, there is a white section with the heading 'Copy, paste... Bot!'. Below this heading, it says 'Create bot dialogue as easily as pasting in a URL or filling in a table. Now anyone on your team can be part of the bot development process.' To the right of this text is an icon showing two overlapping document pages with a plus sign between them, and a clipboard icon to the right. At the bottom left of the white section, there is a text input field containing 'qnamaker.ai/Create' and a blue 'RE >' button.

IBM Tone Analyzer

IBM Watson Developer Cloud
Services Docs Starter Kits Community



Tone Analyzer

This service uses linguistic analysis to detect joy, fear, sadness, anger, analytical, confident and tentative tones found in text.

Resources:
[Documentation](#)
[API Reference](#)
[Fork on Github](#)

Start free in Bluemix

Sample use cases

Choose an example to learn how you can adjust the tone of your content to change people's perceptions, or improve its effectiveness. [Learn more](#).

☒ Tweets
☐ Online Review
☐ Email message
☐ Product Review in French
☐ Your own text

Analyzing Customer Engagement Data? Try out the [Tone Analyzer Customer Engagement Endpoint](#).

I hate #ThisPhoneCompany products, you'd have to torture me to get me to use #thisphone.

The emojis in #ThisPhone are stupid.

#ThisPhone is a useless, stupid waste of money.

#ThisPhone is the worst phone I've ever had - ever 😡.

#ThisPhone another ripoff, lost all respect SHAME.

I'm worried my #ThisPhone is going to overheat like my brother's did.

#ThisPhoneCompany really let me down... my new phone won't even turn on.

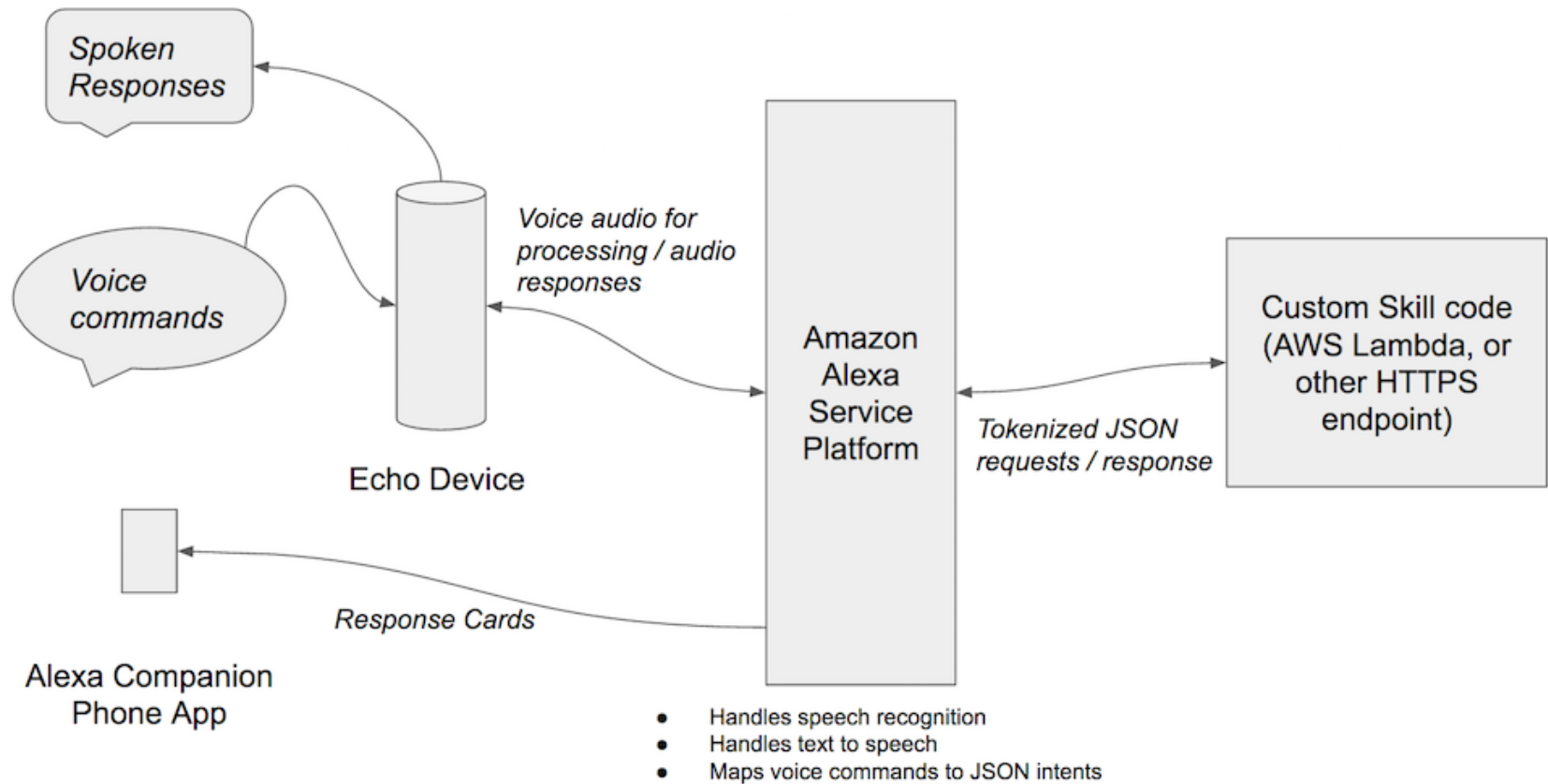
Analyze

Alexa (A), Siri (S), Google Assistant (GA), Cortana (C)

Feature	Best		Worst	
	Correctness	Naturality	Correctness	Naturality
Shopping and Buying Assistant				
Introductions	S	GA	C	A
Managing shopping lists	C	A	GA	GA
Shopping online	A	A	S, C, GA	S, C, GA
Finding restaurants	C, GA	C	A	S
Store schedules	C	C	S	GA
Travel and Entertainment Assistant				
Sport updates	A, S	S, C	GA	GA
Movie info	C	S, GA	A	C
Transportation info	S	A	GA	C
Transportation time	GA	GA	A	A, S
Identifying a song	C	C	GA	GA
Administrative Assistant				
Alarms and timers	A, S, GA	A, S	C	C
Managing to-do lists	A	A	GA	GA
Reminders	S, C	S	GA	GA
Reminders in a date	C	C	GA	A
Schedule a meeting	C	C	GA	GA
Compose an email	S	S	A	A
Miscellaneous				
Local news	C, GA	GA	S	S
Traffic updates	C	C	GA	GA
Weather forecast	A, S, C, GA	S	N/A	C

Forrás: Gustavo López et. al.: Alexa vs. Siri vs. Cortana vs. Google Assistant: A Comparison of Speech-based Natural User Interfaces

Amazon Alexa



Alexa: Skill (Trigger word), Intent (Utterance) és Slot

- **Skill:** képesség valamire

pl. tények lekérdezése, otthonvezérlés, időjárás, mozi, pizzarendelés, viccmesélés

Számos beépített képesség, az alkalmazásboltból tovább bővíthető.

Saját képességeket is fejleszthetünk.

- **Trigger word / Invocation name:** a képességet aktiváló **hívószó**

„Alexa” - az általános hívószó, ami aktiválja a rendszert

„radio” - internetes rádióállomások lejátszása

- **Intent:** a felhasználó **szándéka**

Mit szeretne végrehajtani. Milyen backend API funkciót aktiváljon a rendszer.

- **Sample utterances:** azok a kifejezések, amelyekkel felismerhető az Intent

„Alexa, turn on the radio”, „Alexa, play my last radio station”

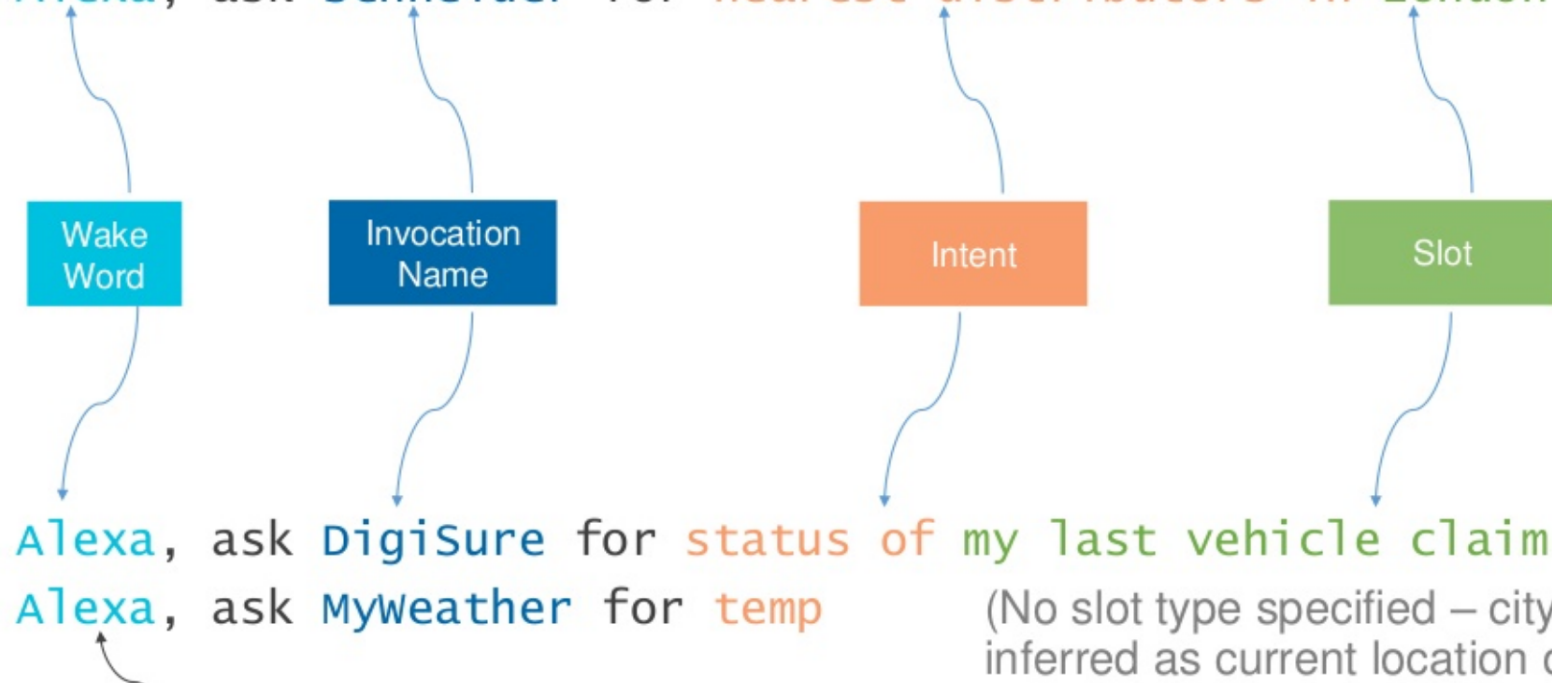
- **Slot:** az Intent **argumentuma**, amely szerepelhet a kifejezésekben

pl. melyik rádióállomást szeretné hallgatni a felhasználó

„Alexa, play {radio station}”, „Alexa, what is the **weather** in **Budapest today**?”

Alexa példák

Alexa, ask Schneider for nearest distributors in London



Wake Word could be configured to
Alexa, Amazon or Echo

Próbáljuk ki az Amazon Alexát!

- 1. Regisztráció Amazon fejlesztőként
http://developer.amazon.com/alexa - Sign in - Create account
(AWS regisztráció nem szükséges.)
- 2. ASK toolkit telepítése
pl. Visual Studio Code + ASK toolkit extension
F1 – ask – ...
- 3. ASK toolkit inicializáció
 - ask init
- 4. Saját skill fejlesztése
 - Invocation name: legyen egyedi (pl. keresztnév + egy szám)
 - Intent: elsőként egy egyszerű „hello world” jellegű
- 5. Skill végpont fejlesztése
 - kipróbálható az Amazon Lambda
 - fejleszthető saját végpont (**JSON API**)
 - használható a tanszéki demo gateway: <https://semcop.mit.bme.hu/api/numbers/>
- 6. Skill tesztelése
 - Alexa mobil app
 - echosim.io

Tervezzünk nyelveket!

- Természetes nyelvek helyett...
 - időben és térben is változnak
 - túl nehéz az értelmezés (nagy háttértudás)
- ... tervezzünk célorientált, korlátos nyelveket!
 - rögzítsük a nyelvtani szabályokat és a szókincset
 - tegyük egyértelművé a közlések értelmezését
- Sajnos új problémákat generáltunk
 - Hogyan hozzuk létre a nyelvet?
 - Milyen nyelvtani konstrukciókat engedünk meg?
 - Milyen elemzőt használjunk?
 - Mekkora legyen a szókincs?

 - Hogyan vegyük rá a felhasználót a szabályok betartására?
 - Honnan tudja a felhasználó, mi része a szókincsnek?

Kontrollált természetes nyelv

- A természetesre hasonlító mesterséges nyelv
 - a sikeres számítógépes feldolgozhatóság érdekében
 - szűkíti egy nyelv nyelvtani szabályait, szókincsét és szemantikáját
 - megőrizve annak természetes jellegét.
- Példák
 - [Basic English](#) (1932), [EasyEnglish](#) (1997), [Wikipedia Simple English](#)
 - INTELLECT natural language database query system (1981), ...
 - Caterpillar, Airbus, Boeing, ... technical languages (1973-), 40+ féle
 - FAA Air Traffic Control (2010), [AECMA/ASD technical english](#) (1980)
 - [Attempto Controlled English](#) (ACE, 1995, 2008), [CLCE](#) (Sowa, 2004)
 - CLOnE (2007), GINO (2006), Ginseng (2006)
 - ... ezernyi más, l. „[Kuhn: Survey and Classification of CNLs](#)” + ábra

Attempto Controlled English (ACE)

- Az angol nyelv egy részhalmaza
 - egyes- és többesszám kezelése (a man, some men)
 - egzisztenciális és univerzális kvantorok (there is a man, every man)
 - számosság (5 men, at least 5 men, less than 5 men)
 - névmások, vonatkozó és általános is (a man that has a cat, everybody)
 - negálás, konjunkció, diszjunkció, utaló hivatkozások főnevekere, stb.
- A mondatok szerkezete
 - Deklaratív mondatok (Every man has a dog.)
 - Változókkal kiegészítve (There is a man X who has a dog.)
 - Eldöntető kérdések (Does Peter have a dog?)
 - Wh-kérdések (Who has a dog? Peter has what?)
- Elsőrendű logikai állításokra képezhetők le a mondatai
 - Az eszközkészletében található elemző (APE) végzi el a leképezést.
 - A következtető gépe képes így kialakított tudásbázisok használatára
- Kipróbálható: <http://attempto.ifi.uzh.ch/ape/>
- Integrálható: <http://technologies.kmi.open.ac.uk/aqualog/>

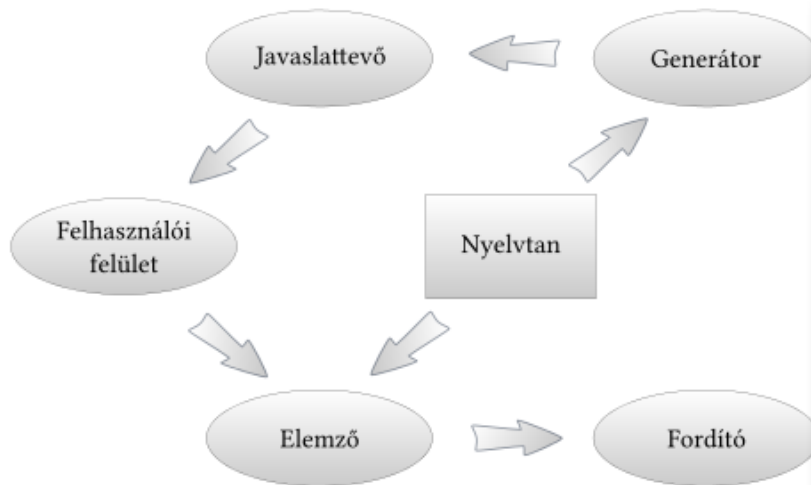
Ontológia szerkesztés: CLOnE, GINO

- A szemantikus publikálás megköveteli ontológiák kidolgozását
 - Az RDF/OWL egyre szélesebb alkalmazói körrel bír, de
 - a szerkesztőeszközök (pl. Protege) használata nem triviális.
- **A kontrollált nyelvek** (Rabbit, CLCE, ACE-OWL, PENG-OWL, SOS,...)
 - lehetővé tehetik laikus felhasználók számára is az ontologiaszerkesztést
 - használatuk jóval kevesebb betanulást igényel, mint egy OWL szerkesztő
- CLOnE: Controlled Language for Ontology Editing (2007)
 - A **GATE keretrendszert** használja a mondatok elemzésére
 - Nyelvi elemzője laza: kulcs kifejezéseket próbál megtalálni és értelmezni
- GINO: Guided Input Natural Language Ontology Editor (2006)
 - alkalmas lekérdezésre (**SPARQL** nyelvre fordít), és szerkesztésre is
 - inkrementális nyelvi elemző segíti a kontrollált nyelv használatát
 - a **Jena Semanti Web** keretrendszert használja

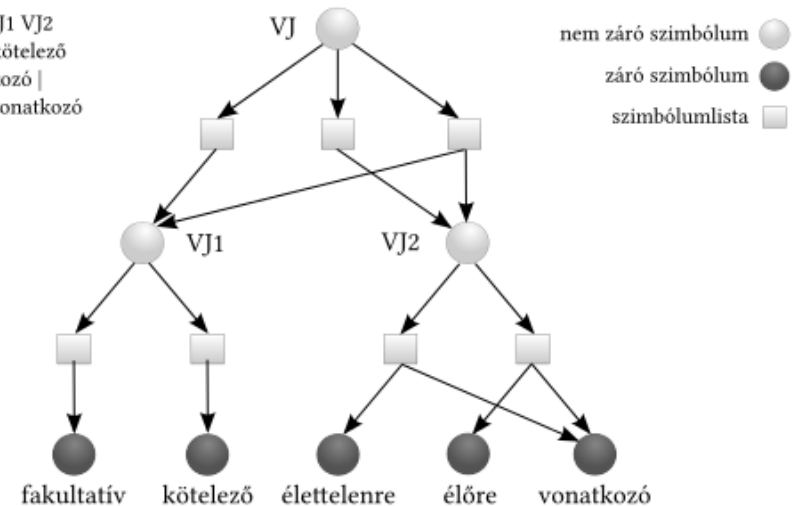
Kontrollált nyelvű lekérdezés

- A természetes és a formális nyelvű lekérdezők közé épít hidat
 - A felhasználók által könnyebben elsajátítható, és
 - a formális lekérdező nyelvekre egyértelműen lefordítható
- Adatbázisok lekérdezése („a klasszikus”)
 - Általános természetes nyelvű lekérdezőként indult a 60-as években
 - Kommerciális termékek is megjelentek:
INTELLECT, IBM LANGUAGEACCESS, stb. (10+)
 - Nem kontrollált, vagy nem pontosan ismert nyelvek
- Ontológiák lekérdezése
 - A szerkesztők „mellékterméke”, folytatása, előzménye
 - QuestIO (2008): a CLOnE lekérdező rendszere, SPARQL-re fordít
 - A GINO rendszer lekérdezésre is alkalmas
- Tudástárak lekérdezése jellemzően *alany-reláció-tárgy* kapcsolatokra
 - Általános célú, logikai-alapú nyelvek környékén (ACE, CLCE, stb.)
 - Pl.: AquaLog (2004), FREyA (2010)
 - GAPP (2003): „Foundational Model of Anatomy” tudásbázis lekérdezése
 - ...

Esettanulmány: adatbázis-lekérdezés



$VJ = VJ1 \mid VJ2 \mid VJ1 VJ2$
 $VJ1 = \text{fakultatív} \mid \text{kötelező}$
 $VJ2 = \text{éltlenre vonatkozó} \mid \text{éltlenre vonatkozó}$



Főnévi vonzattár

Felhasználó: Mészáros Tamás [Portál admin] (Kijelentkezés)
Nyomtatható változat

[Információk](#)

[Keresés főnevek és vonzatok szerint](#)

[Publikációk](#)

Navigáció: [Főnévi vonzattár - kérdés a főnévlistához](#)

Keresés a főnévi vonzattárban kérdés segítségével

Az alábbi mezőben magyar nyelven lehet feltenni főnevekkel és vonzataikkal kapcsolatos kérdéseket. A rendszer csak bizonyos szerkezetű kérdéseket tud megválaszolni, ezért kontrollálja a kérdés feltevését. Első használat előtt érdemes elolvasni a [prediktív szövegbevitellel kapcsolatos tájékoztatót](#).

Kérdés:

melyik

melyik

DEBUG: ☐

A kontrollált nyelvű bevitel támogatása

- Problémák a kontrollált nyelv alkalmazása során
 - Hogyan vegyük rá a felhasználót a szabályok betartására?
 - Honnan tudja a felhasználó, mi része a szókinsznek?
- Lehetséges megoldások
 - Megtanítjuk a felhasználót a szabályokra és a szókinszre
 - A bevitel során nincs támogatás, de az elemzés korrigál, visszajelez
 - Valamilyen beviteli támogatást alkalmazunk
- A szövegbevitel támogatása
 - Az egyszerű menüalapú kiválasztástól
 - a mondatban szereplő fogalmak, konstrukciók grafikus kiválasztásán át
 - egy szövegrészlet lehetséges bővítéseit dinamikusan generáló rendszerig
 - Ez utóbbi új követelményeket támaszt a nyelvi elemzővel szemben.