

Szöveg- és Webbányászat házi feladat beszámoló

Távközlési és Médiainformatikai Tanszék

|  |  |
| --- | --- |
| Készítette: | **Szántó Tamás, Benda Krisztián** |
| Neptun-kód: | **ET7D8H, J1CEI3** |
| Ágazat: | **Adat- és Médiainformatika** |
| E-mail cím: | [**tmas.szanto@gmail.com**](mailto:tmas.szanto@gmail.com)**,** [**krisztianbenda@gmail.com**](mailto:krisztianbenda@gmail.com) |
| Konzulens(ek): | **Dr. Szűcs Gábor** |
| E-mail címe(ik): | **szucs@tmit.bme.hu** |

Téma címe: Névelem felismerés

Feladat

A névelem-felismerést (named entity recognition) segítségével kinyerhetők egy adott korpuszon belül előforduló névelemek, s ezen belül a tulajdonnevek (személynevek, helyek, szervezetek és egyéb tulajdonnevek). A feladat angol nyelvű szövegben 7 típusú entitásnak a felismerése, melyek a következők:

event = esemény; geo = földrajzi entitás; gpe = geopolitikai entitás; obj = objektum, műtárgy; org = szervezet; per = személy; time = idő

A felismerendő entitások állhatnak 1 vagy akár több szóból is. Minden esetben az entitás első szavát külön detektálni kell, ennek jelzésére a B betű használandó (beginning): így B-event, B-geo, stb. címkékkel kell a megfelelő szavakat ellátni. Ha az entitások több szóból áll, akkor az összes többi I-vel jelölendő (inside), azaz I-event, I-geo, stb. címkék; így összesen az egyéb (O) címkével együtt 15 osztálycímke adódik.

Példa:

Indian border security forces are accusing their Pakistani counterparts of lobbing at least four rockets into northern Punjab state.

Indian: B-gpe, Pakistani: B-gpe, Punjab: B-geo, a többi pedig O címkéjű.

**2018/2019. 1. félév**

# Megoldási terv

Félév elején a házi feladat megoldási tervünket a következőképpen foglaltuk össze:

Vállalt részfeladatok:

1. Létező megoldások vizsgálata és kipróbálása
2. Órán tanult módszerek áttanulmányozása
3. A tapasztalatok alapján prototípus elkészítése
4. Az elkészült megoldás javítása, továbbfejlesztése
5. Bemutató elkészítése és előadása

Megoldási ötletek:

* Az általánosabb feldolgozási folyamat a következő:
  + Tokenizálás, normalizálás/szótövezés, névelem detektálás, névelem normalizálás
* Névelem detektálására az alábbi megközelítéseket ismerjük
  + Szótár alapú
    - Összes entitás összes formáját össze kell gyűjteni
    - Nagy tudásbázis vagy annotált korpusz szükséges
  + Szabály alapú
    - Mintákat kell írni az entitások illesztéséhez
    - Téma specifikus tudás szükséges
  + Statisztikai modell alapú
    - Valószínűségek hozzárendelése a szövegrészekhez
    - Sok tanuló példány szükséges
    - Előny: téma független tudás
* Ezen detekciók használatának előnyeit fogjuk felmérni és ezalapján a legmegfelelőbbet kiválasztani.

Létező eszközök, módszerek:

* [spaCy](https://spacy.io): python library, statisztikai modell alapú NER, sokféle kategória támogatott, továbbtanítható saját kategóriákkal
* [Stanford NER is a Named Entity Recognizer](https://nlp.stanford.edu/software/CRF-NER.shtml): Java library, kevés alapból támogatott kategória
* [Named-Entity-Recognition-BLSTM-CNN-CoNLL](https://github.com/mxhofer/Named-Entity-Recognition-BidirectionalLSTM-CNN-CoNLL): implementáció [ehhez a cikkhez](https://arxiv.org/abs/1511.08308), Keras

Használni tervezett technológiák:

* Elsősorban Python 3-at szeretnénk használni
* Kisebb részfeladatok/algoritmusok kipróbálásához opcionálisan RapidMinder-t is igénybe vennénk
* A párhuzamos munkavégzést a GitHub segítségével oldanánk meg

# **Névelem felismerés megoldása**

A féléves munkánkat az alábbi négy részre lehet osztani:

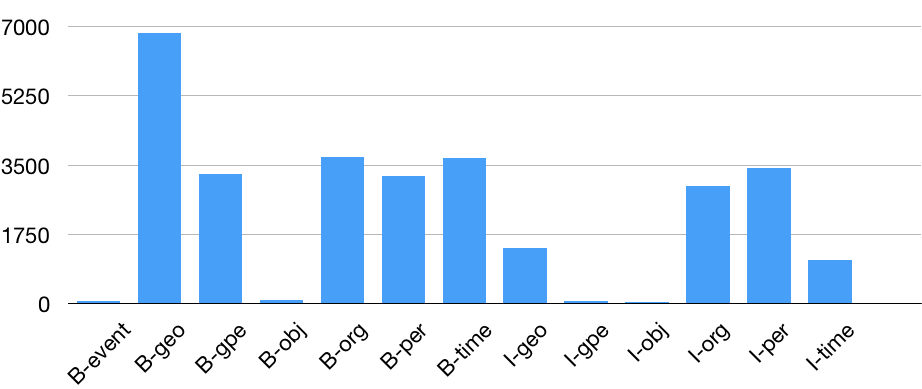
* Feladat megértése, adathalmaz tanulmányozása
* Névelem felismerő rendszer kiválasztása
* SpaCy tanításának és használatának kipróbálása első megoldás elkészítése
* Hosszabb tanítások kipróbálása és az eredmény javítása

A beszámolónkat is ebben a sorrendben készítettük el, kiegészítve egy összefoglalóval.

## **2.1. Feladat megértése, adathalmaz tanulmányozása**

A hivatalos házi feladat leírását már ismertettük a címlap Feladat résznél, azonban egy adatbányászati problémához és technikák kiválasztásához mindenképp szükség van az adathalmaz struktúrájának megértéséhez. A házi feladathoz három adathalmaz állt rendelkezésre: TrainNER, Test1NER és Test2NER. A TrainNER négy, a Test1NER három oszlopból áll, míg a Test2NER kettőből. Mindhárom esetben kerek mondatokból álló szöveget tartalmaznak az adathalmazok, és az első oszlop a mondathatárokat azonosítja, míg a második oszlopban a szavak találhatók meg. A TrainNER és Test1NER kibővül egy POS tageket tartalmazó oszloppal, illetve értelemszerűen a TrainNER tartalmazza soronként az elvárt névelemeket, amelyek a feladatleírásban is szerepeltek.

Mivel sem a Test1NER-hez sem a Test2NER-hez nem rendelkeztünk helyes névelemekkel, ezért a TrainNER-t vetettük alaposabb vizsgálat alá. A TrainNER 9000 mondatot tartalmaz és 196645 szót összesen 14 névelem szerepel benne, tehát egy a feladatban ismertetett névelem teljesen hiányzik (I-event). A következő grafikonon vizualizáltuk a névelemek eloszlását az adathalmazban (1. ábra).

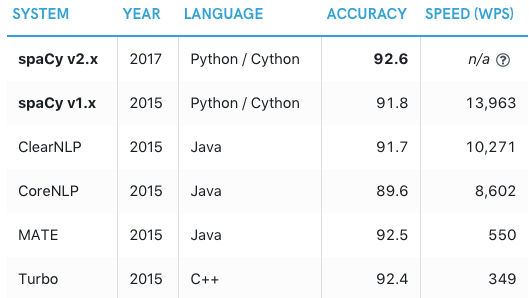


1. ábra: Névelemek eloszlása a TrainNER-ben

Az „*O”,* mint *Other* típusú névelemet nem jelenítettük meg az ábrán, mert nagyságrenddel több esetben fordul elő, mint a többi névelem. Összesen 166 610-szer, ami az adathalmaz közel 85%-a. Tehát, csak „*O”* predikcióval már egész magas pontosságot lehet elérni.

## **Névelem felismerő rendszer kiválasztása**

A tárgy keretein belül RapidMinert és Python-t használtunk, mivel személyesen is szeretjük a Python nyelvet ezért házifeladatunkat is elsősorban ezen a nyelven képzeltük el. Internetes keresés után rábukkantunk a SpaCy [1] nevezetű természetes nyelv feldolgozó megoldásra. A SpaCy többfajta megoldást kínál természetes nyelv feldolgozásához. Többek elérhető vele névelem felismerés is. A jelenlegi megoldásokkal összehasonlítva is elég jó helyen szerepel, amely a következő ábrán látszódik (2. ábra):



2. ábra: Természetes nyelvfeldolgozó megoldások összehasonlítása [2]

Bár a táblázat általánosságban hasonlítja össze a nyelvfeldolgozókat, NER megoldás esetén is jónak mondható a SpaCy. További pozitívumot jelentett, hogy a SpaCy alkalmas tanításra is és előre megalkotott modelleket is biztosít:

|  |  |
| --- | --- |
| Név | Méret |
| *en\_core\_web\_sm* | *35 MB* |
| *en\_core\_web\_md* | 115 MB |
| *en\_core\_web\_lg* | 812 MB |

## **SpaCy tanításának és használatának kipróbálása első megoldás elkészítése**

# Hivatkozások

1. SpaCy szoftvermegoldás, <https://spacy.io> (2018. 11. 26.)
2. Természetes nyelvfeldolgozó megoldások összehasonlítása, <https://spacy.io/usage/facts-figures#benchmarks> (2018.11.26)