Entitásorientált véleménykinyerés magyar nyelven

Huszti Dániel és Ács Judit

Budapesti Műszaki és Gazdaságtudományi Egyetem,
Automatizálási és Alkalmazott Informatikai Tanszék,
huszti.daniel@gmail.com, judit@aut.bme.hu

Kivonat Napjainkban a digitális formában fellelhető, strukturálatlan adatok mennyisége folyamatosan növekszik, ezáltal a bennük említett entitásokra vonatkozó vélemények polaritásának automatizált elemzése is egyre fontosabbá válik. Cikkünkben bemutatunk egy olyan alkalmazást, mely segítségével magyar nyelvű szövegekből lehetséges a tulajdon-, földrajzi- és cégnevekre vonatkozó, részletes szerzői attitűd kinyerése. A forráskódot és a megoldást virtualizált formában is nyilvánosságra hoztuk.
Kulcsszavak: véleménykinyerés, polaritás, szentiment, természetesnyelv-feldolgozás

1. Bevezetés

Az információs társadalom által generált szöveges adatok mennyiségének drasztikus növekedésének köszönhetően az automatizált elemzési megoldások egyre szélesebb körben kezdtek elterjedni. Ezen igaz a véleménykinyerés területére is, a szövegrészletekben előforduló különböző entitásokra (tulajdonnevek, földrajzi és cégnevek) lebontva, részletes kimutatások előállítására mutatkozik jelentős piaci igény.

Magyar nyelvre publikusan elérhető szentiment korpuszok száma csekély, ezek közül entitás-szintű véleménykinyerésre egyedül az OpinHuBank alkalmas. Munkánk során ez utóbbi korpuszt felhasználva úgy tanítottuk be a modellünket, hogy képes legyen az egyes entitásokra vonatkozó polaritások megállapítására. Az implementáció során törekedtünk a valós életben történő alkalmazhatóságra, ezért az iteratív fejlesztési folyamat során valós példákon is megvizsgáltuk az éppen aktuális modell pontosságát. A nyilvánosságra hozott alkalmazásban moduláris felépítést alkalmaztunk a továbbfejleszthetőség érdekében. Az intuitív módon használható fejlesztői interfész és a Docker container technológia által garantált platformfüggetlen futtathatóság nagyban segítheti az applikáció felhasználását.

2. Létező megvalósítások

A természetes nyelvfeldolgozás, azon belül a véleménykinyerés napjaink egyik legnépszerűbb kutatási területévé emelkedett, melyet a nemzetközi versenyekre és konferenciákra benyújtott számos koncepción felül a nagyvállalati megoldások jelenléte is alátámaszt. Utóbbira jó példa a világ egyik legnagyobb videostreaming szolgáltatója, mely valós időben történő véleménydetektáló rendszer integrálásával biztosít interaktív videózási élményt.

Az egyik legjelentősebb megmérettetés az Association for Computational Linguistics (röviden ACL) intézet által szervezett SemEval [1] [2] [3], amely

évről-évre egyre több jelentkezőt vonz, akik több különféle komplexitású feladatban is összemérhetik megoldásaik hatékonyságát. Az utóbbi három évben egyre nagyobb jelentőséget kapott a véleménykinyerés szekció, azon belül pedig az aspektus-szintű elemzés, eleinte csak mondatszintű, majd szövegrészletre kiterjesztett, 2016-ban pedig már domainen túlívelő szentiment analízis feladatok is kitűzésre kerültek.

A mondatszintű véleménydetekciós megmérettetés alapja az elmúlt három évben változatlan, az éttermekre és laptopokra vonatkozó értékelésekből az aspektusokhoz (pl. étel vagy kiszolgálás minősége) tartozó vélemények kategóriájának definiálása a cél. Megvizsgáltuk az ott részletezett koncepciókat, a legjobbak háromosztályos aspektus-szintű szentiment elemzésre 80% feletti pontosságot tudtak elérni. Ezen megoldások alapkoncepciója majdnem minden esetben azonos, az alapvető nyelvi elemzés eszközei segítségével mondat- és szóhatárok, szófaj és morfológiai felbontás meghatározását, majd a funkciószavak kiszűrését követően a szótövezett alakokra unigram, néhol bigram jellemzők illetve feladatspecifikus súlyozás vagy dimenziócsökkentés kerül alkalmazásra. Utóbbi kettő a polaritás szempontjából érdekes szavak kiemelésére használatos technika. Általánosságban vett optimális megoldás nem létezik, mivel gyakran egyedi, a korpuszra jellemző tulajdonságokat vesznek figyelembe.

Megvizsgáltunk egy cseh nyelvre elkészített megoldást is [4], melyben uni- és bigram jellemzők szótövezett és az eredeti alakját jegyként felhasználva 66,27%-os pontosságot értek el háromosztályos aspektus-szintű szentiment elemzésre.

A korszellemnek megfelelően számos neurális hálós megoldás is született entitásorientált véleménykinyerésre[5,6].

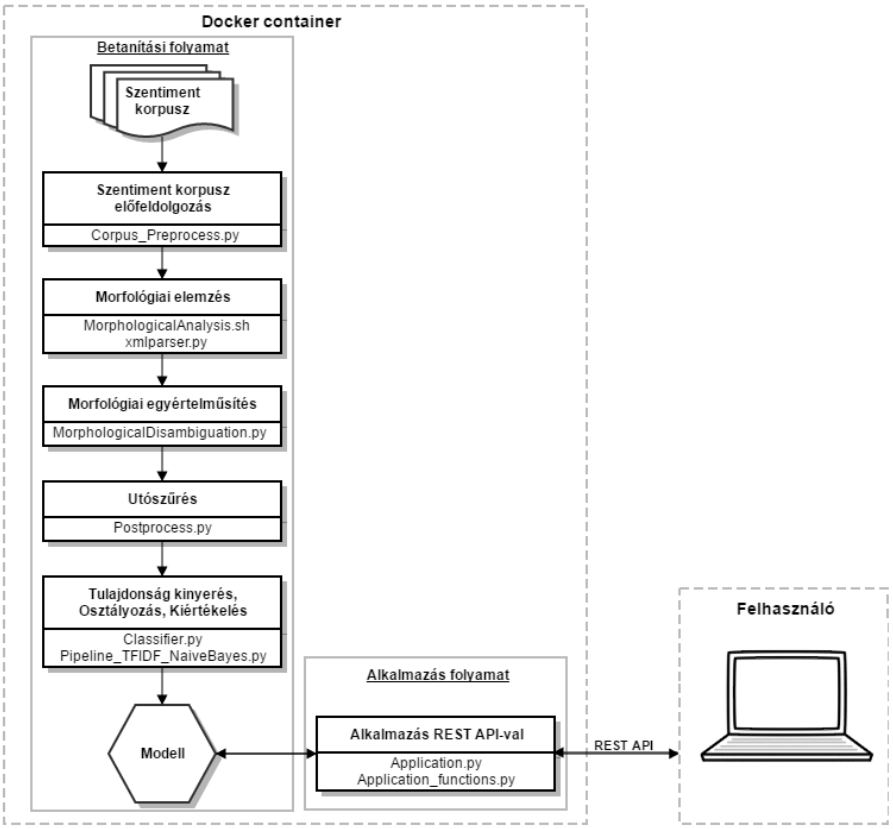
Magyar nyelven talán a *Trendminer* [7,8] a legismertebb megoldás, amely az OpinHuBank szentiment korpuszon uni- és bigram jellemzők felhasználásán felül speciális, távolság alapú súlyozás illetve polaritáslexikonok segítségével három- és kétosztályos esetben is 80% feletti pontosságot ér el.

Az imént említett magyar nyelvre implementált megoldások forráskódját nem hozták nyilvánosságra, ezért úgy gondoltuk, hogy érdemes egy a SemEval aspektus-szintű véleménykinyerés feladatához, és a [9] cikkhez hasonló, nyílt forráskódú alkalmazást elkészíteni, mely képes a szabad entitásokhoz kapcsolódó vélemények kategorizálására. Mivel a magyar nyelvre elkészített, publikusan elérhető szentiment korpuszok száma nagyon korlátozott, ezért az egyetlen, ilyen entitás-mondat párokat tartalmazóra, az OpinHuBank adatbázisra [10] esett a választásunk. A főként internetes hírportálokról, blogokról letöltött mondatokat öt természetes személy annotálta pozitív/semleges/negatív kategóriák egyikébe. Az entitás jelen esetben mindenképpen egy természetes személy, azonban ez attól még jó alapként szolgálhat a modell későbbi általánosítása céljára, így az kis módosítással akár termékekről keletkező vélemények elemzésére is alkalmassá válhat.

Tudomásunk szerint az egyetlen szabadon elérhető magyar nyelvű véleménykinyerő a Polyglot sentiment analysis modulja [11], ami támogatja a magyar nyelvet is, amellyel össze is hasonlítottuk a mi megoldásunkat.

3. Alkalmazott módszerek

Az alkalmazásunk felépítését a 1 ábra szemlélteti. Alapvetően három részre bontható: egy előfeldolgozó, egy nyelvfeldolgozó vagy NLP és egy gépi tanuló modulra.

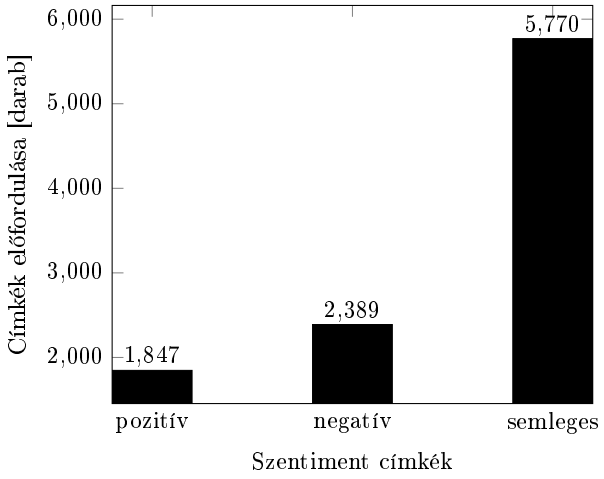


1. ábra. Alkalmazás felépítése

A nyelvfeldolgozó modul megvalósításához a BME MOKK Hun* eszközeit, a PrecoSenti magyar szentiment lexikonjait[1], a Polyglot NER tulajdonnév detektálásra szolgáló eszközét [12] alkalmaztuk, míg a gépi tanuló modulhoz a Python Sklearn [13] csomagját használtuk fel. A betanítás előtt a tanító és teszt adathalmaz 80-20% arányban történő véletlenszerű szétválasztását alkalmaztuk. A tulajdonságkinyerést és a tanító algoritmus futtatását az Sklearn Pipeline segítségével automatizálva végeztük. Az optimális paraméterek kiválasztását hasonlóképp az Sklearn GridSearchCV funkciója segítségével, tízszeres keresztvalidációval határoztuk meg.

3.1. Korpusz előkészítése

Az OpinHubank 5 annotátor értékeléseit tartalmazza, akik közt az egyetértés nagyon változó. Az annotátorok által adott pontszámokat összeadtuk, azonban így is az entitások 57,76%-a kapott semleges értékelést, míg negatív (-5– -1), illetve pozitív (1 – 5) értékelést egy pontszámra levetítve nagyon kevés entitás kap (200–500 kategóriánként). A negatív és pozitív pontszámokat egyetlen negatív, illetve pozitív kategóriára vetítettük le, ezáltal három osztályt hoztunk létre (negatív, semleges, pozitív). Az osztályok eloszlását a 2 ábra szemlélteti.



2. ábra. OpinHuBank mondatainak kategorizálása

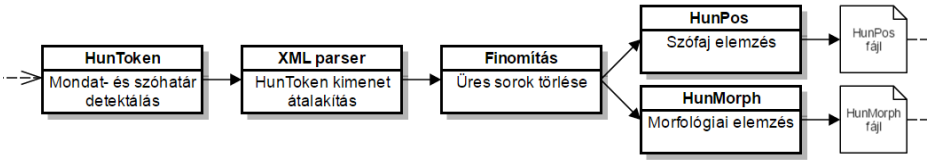
Az optimális megoldás megtalálása érdekében elvégeztem a három – pozitív/semleges/negatív – kategóriára történő szűkítést, azaz az előző kalkulált érték alapján nullánál nagyobb értékkel szereplő előfordulásokat pozitív, a kisebbeket pedig negatív címkével láttuk el. Ezen kategóriák aránya látható a fenti ábrán.

A korpusz automatizált feldolgozása érdekében még kisebb adatmanipulációs műveleteket is szükséges volt elvégeznünk annak érdekében, hogy a feldolgozásra kialakított pipeline megfelelően tudjon működni. A mondat végén alkalmazott rövidítések gyakran rossz döntésre késztették a mondathatár elválasztásért felelős HunToken eszközt, ezért hozzáadtunk „.”-ot a mondat végéhez. Hasonló problémával szembesültünk, amennyiben a mondat első karaktere kisbetűs volt, ezért azokat nagybetűssé alakítottuk.

[1] http://www.opendata.hu/storage/f/2016-06-06T11%3A27%3A11.366Z/precosenti.zip

3.2. Morfológiai elemzés és egyértelműsítés

A nyelvfeldolgozó pipeline-t a 3 ábra szemlélteti. Tokenizáláshoz a HunTokent használjuk, amelynek xml kimenetét plain textté alakítjuk és az üres sorok eltávolítása után a HunPos [14], illetve a Hunmorph [15] segítségével szófaji és morfológiai elemzést végzünk. A morfológiai egyértelműsítést a szófaji címkéket felhasználó heurisztika alapján végezzük.



3. ábra. Morfológiai elemzés folyamata

Amíg a HunPos figyelembe veszi annak mondatbeli kontextusát, ezért egyértelmű értéket rendel minden egyes szóhoz, addig a HunMorph az egyes szavak összes lehetséges morfológiai felbontását adja kimenetül, ezért egy morfológiai egyértelműsítő implementálására volt szükség. Utóbbi megvizsgálja, hogy hány és milyen kimenettel rendelkezik a HunMorph, majd kiválasztja a HunPos szófajának megfelelő kimenetűt. Előfordulhat, hogy több megoldás is létezik, ilyenkor az elsőt választjuk. Az így előállított kimeneten egy paraméter segítségével állítható, hogy a szótövet vagy a szófajt is tartalmazó alakot használjuk fel betanításra.

3.3. Utószűrés

Az előkészített tokeneken már elvégezhető lenne az elemzés, azonban előtte még kisebb utószűrési feladatok elvégzését láttuk célszerűnek. A funkciószavak (stopwords) szűrésen felül, a számok tanító halmazból történő eltávolítása is hasznosnak bizonyult a szentiment elemzés szempontjából. Továbbá a nagyon ritkán – jelen esetben háromnál kevesebbszer – előforduló kifejezések egy új, eddig nem létező tokennel kerültek helyettesítésre.

3.4. Tulajdonságkinyerés és felügyelt gépi tanulás

A rendelkezésre álló adatok ritkaságát figyelembe véve és a számítási kapacitás csökkentése érdekében úgy döntöttünk, hogy a következő, kizárólag unigram alapú tulajdonságokat fogjuk alkalmazni a modell betanítása során:

Szimmetrikus n szó széles ablak. A korpuszban több olyan mondat is szerepel, melyben több különböző entitást is tartalmaz, ezért azok környezetét kiemelt jelentőséggel kezeltük a betanítás során. Ennek érdekében egy entitás körüli szimmetrikus n széles ablak alkalmazása mellett döntöttünk. A legjobb konfiguráció 5 széles kontextust vesz figyelembe a szó előtt és után is.

Szavak előfordulása TFIDF szerint súlyozva. A szentiment elemzés során nagyon gyakran alkalmazott módszer, mellyel a gyakran előforduló szavak kicsi, míg a ritkábban előforduló kifejezések magasabb súllyal vesszük számításba. Ezáltal a véleménykinyerés szempontjából fontos kifejezéseket magasabb értékkel szerepeltetjük. Erre a célra az Sklearn TFIDFTransformer függvényét használtuk lineáris TF és smooth IDF paraméterekkel. Utóbbit a nullával történő osztás elkerülése végett alkalmaztuk.

Szentiment szótárakban előforduló szavak száma. A modell pontosságának javításán túl a valós életben történő használhatóságot is figyelembe véve, célravezetőnek véltük előre elkészített szentiment szótárak alkalmazását. A PrecoSenti pozitív és negatív polaritáslexikonok külön-külön tulajdonságként kerültek implementálásra, s feleakkora súllyal lettek figyelembe véve. Előbbi 1748, utóbbi 5940 kifejezést tartalmaz.

Az optimális modell elkészítése érdekében több különféle osztályozó algoritmust is kipróbáltunk, mint az SVM több különböző kernellel, multinomiális NaiveBayes és a logisztikus regresszió. Az optimális paraméterek megválasztását is automatizálva végeztük, az Sklearn GridSearchCV funkció segítségével, tízszeres keresztvalidációval. Az elkészített modell a független teszthalmazon került kiértékelésre.

4. Eredmények

A feladat megvalósítása kapcsán arra törekedtünk, hogy a rendszer ne csak az arany sztenderként alkalmazott OpinHuBank korpuszon, hanem lehetőleg valós körülmények között is alkalmazható legyen, így a modell fejlesztése és tesztelése kapcsán több valós példán is tesztelést végeztünk.

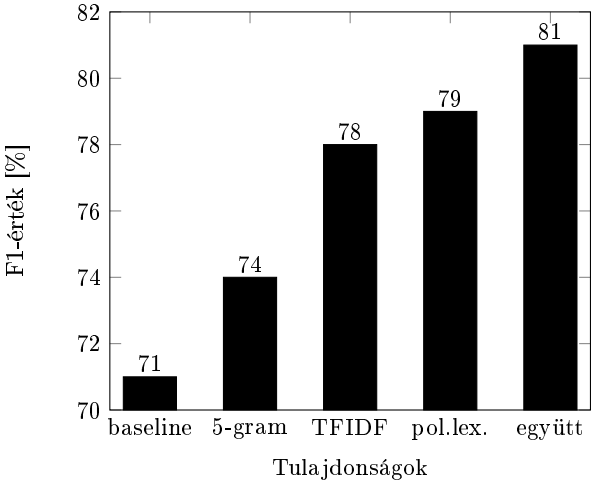
4.1. Diszkusszió

Mivel a szövegbányászati modell fejlesztése iteratív feladatnak számít, az ideálisnak vélt tulajdonságok, gépi tanuló algoritmus és paraméterek kiválasztását csak több teszt futtatása után tudtuk meghatározni. A kiértékeléshez pontosságot (precision), fedést (recall) és F1-mértéket (F1 score) használtunk.

Első körben egy háromosztályos véleménydetekciót végeztünk, azonban a kiértékelés során mért 66% feletti F1-mérték ellenére, a túl nagy számban jelenlévő semleges vélemények miatt a valós életbeli példákon nagyfokú torzítás jelentkezett. Emiatt a semleges teszthalmaz eltávolítása mellett döntöttünk, s ilyen módon is részletes vizsgálat alá vetettük az eredményeket. Ezúttal azt tapasztaltuk, hogy a modell kiértékelése során kapott F1-mérték nagyjából megegyezik a valós életből vett mintapéldákra letesztelt eredményekkel.

A felügyelt gépi tanuló algoritmus közül a kifejezetten szövegelemzési célra fejlesztett multinomiális Naive Bayes alkalmazás bizonyult célravezetőnek, azonban csak minimális, körülbelül 1%-kal jobb eredményt biztosított, mint az SVM algoritmus lineáris kernellel vagy a logisztikus regresszió. Ezzel szemben a korpusz megfelelő előfeldolgozásával, és a tulajdonságkinyerés segítségével jelentős pontosságnövekedést értünk el.

A fenti ábrán szemléltetésre kerültek a kétosztályos esetre elkészített tulajdonságok alkalmazása külön-külön, illetve együttes alkalmazásának hatásai a kiértékelés során mért F1-mértékre. A baseline rendszer, azaz a szimpla unigram



4. ábra. Tulajdonságkinyerés hatása a kétosztályos feladat során

alapú szószámlálást felhasználva elért 71%-os érték jelentősen, 10%-kal növelhető az imént bemutatott jellemzők együttes felhasználásával.

1. táblázat. Legjobb kétosztályos entitásorientált modell kiértékelése

Címke	Pontosság	Fedés	F1-mérték	Teszt bejegyzések száma
negatív	0.78	0.90	0.84	460
pozitív	0.85	0.71	0.77	388
átlag/össz	**0.82**	**0.81**	**0.81**	**848**

A legjobb eredményt fent ismertetett tulajdonságok együttes használata, és a multinomiális Naive Bayes osztályozó következő paraméterei szolgáltatták: alpha: 0.75, class_prior: None, fit_prior: False.

4.2. Eredmények összehasonlítása

Az betanított modell elkészítését követően fontosnak tartottuk annak összehasonlítását meglévő magyar nyelvre implementált megoldásokkal, amelyek kiértékeléssel is rendelkeznek. Az itt bemutatott munkához leginkább a [9] hasonló, amely szintén entitásorientált megközelítést alkalmaz az OpinHuBank korpuszt használja.

A Szegedi Tudományegyetem csapata által elkészített kétosztályos megoldás eredményeihez hasonlítjuk munkánkat. A két koncepció már a korpusz előfeldolgozásánál eltér, mivel ők a nem egyértelmű, azaz a pozitív és negatív értékeléssel is rendelkező entitás-mondat párokat nem vették figyelembe, addig mi az összesített pontszámokat vettük figyelembe és csak a 0 összegűeket dobtuk el.

Az általuk elért legjobb eredmény során kizárólag unigram jellemzőket alkalmaztak, meglepő módon a bigram jellemzők rontottak a modell pontosságán. Tulajdonságként nem csupán az entitások közvetlen környezetét vették figyelembe, hanem azoknak az entitásokhoz vett relatív pozíciója alapján történő súlyozását is. Továbbá alkalmaztak előre elkészített szentiment szótárakat is. Így végül 88,5%-os pontosságot (precision) értek el kétosztályos entitás-orientált szentiment elemzés esetén.

Ugyan a mi legjobb konfigurációnk pontosságban elmarad, azonban szabadon elérhető a forráskód, illetve „dobozos termékként” a Docker image.

A tesztadatokat a Polyglot sentiment analysis magyar moduljával is felcímkéztük. A Polyglot háromféle választ ad: pozitív, negatív és nem meghatározott (cannot determine). A tesztadatok 32%-ára adott nem meghatározott választ, a maradék adaton 69%-os pontosságot ér el, a nem meghatározottakat hibásnak számolva a pontosság csupán 46%.

5. Hibaelemzés

A tesztadatokon végeztünk kézi hibaelemzést, amely során az alábbi hibaosztályokat állapítottuk meg a 154 hibásan osztályozott entitásnál: negálás, kétértelműség, szentiment szótár hibája, adatritkaság (a szótárral nem volt átfedés). A 2. táblázat szemlélteti a hibák gyakoriságát.

2. táblázat. Az egyes hibaosztályok gyakorisága

Hibaosztály	Előfordulás	%
negálás	16	10%
kétértelműség	18	12%
szótár	31	20%
adatritkaság	89	58%

A hibaosztályokat példákkal és magyarázattal szemléltetve:

negálás *Az Országos Igazságszolgáltatási Tanács (OIT) kedden úgy döntött, hogy nem támogatja Baka András főbírói jelölését.*

kétértelműség *Azt azért rendesen röhögöm, hogy a képen minden fordítva van, mint a pártéletben: Fodor a hatalmas, Orbán a törpe, és erre még ráerősít a kép torzítása is.* – az „Orbán a törpe" kétértelmű,

szentiment szótár hibája *A norvég politikusok már feladták a reményt, hogy saját védelmi miniszterük, Kristin Krohn Devold nyerje el a tisztséget.* – a remény pozitív szóként szerepel a szótárban,

adatritkaság *Az ír kormányfő biztosította támogatásáról Orbán Viktort.* – a kormányfő szó nem szerepelt a szótárban, a tanítóadatban többször szerepelt negatív kontextusban.

6. Alkalmazás

A feladat kitűzésekor a valós életben használható modell betanításán felül egy olyan alkalmazás elkészítésére helyeztük a hangsúly, mely bárki számára elérhető, platformfüggetlenül és intuitív módon használható. Előbbi érdekében a telepítő elkészítésén túl létrehoztunk egy előre inicializált Docker containert[2], míg utóbbi érdekében egy REST API hozzáférést nyitottunk.

A Docker image a bemutatott teljes pipeline-t előre telepítve tartalmazza. Az elemezni kívánt szövegrészletet a REST API-n keresztül Windows esetén külön alkalmazásból (példaképp WizTools[3]), Linux vagy Mac OS X operációs rendszernél pedig akár a parancssorból a következőképpen lehet beküldeni:

```
curl -i -H "Content-Type: application/json" -X POST -d
'{"sentence": "Ide írja a az elemezni kívánt szöveget."}'
http://172.0.0.1:5000/sentiment_verbose
```

A predikció során részletes eredményeket közlünk, azaz a teljes szövegrészlet szentimentjén felül az egyes entitásokra kapott pozitív szentiment valószínűségét is megadjuk. Ezek alapján egy harmadik, semleges kategória is kialakításra került, ha a két eredmény között kisebb, mint 15% a különbség. Az entitások és azok kategóriájának (tulajdon, földrajzi és cégnév) meghatározására a Polyglot NER modulját alkalmaztuk, véleményelemzésre pedig az azok körüli szimmetrikus 5 széles kontextusablak alkalmazása után, a szűkített adathalmazon kerül sor. A 5 ábrán egy ilyen részletes elemzésre adunk egy példát.

Az alkalmazás részletes használati útmutatóját, forráskódját, telepítőjét nyilvánosságra hoztuk a GitHubon.[4]

Hivatkozások

1. John Pavlopoulos Haris Papageorgiou Ion Androutsopoulos Suresh Manandhar Maria Pontiki, Dimitrios Galanis. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, SemEval '14, pages 333–352, Dublin, Ireland, 2014. Association for Computational Linguistics.

[2] https://hub.docker.com/r/dhuszti/sentanalysis/
[3] https://github.com/wiztools/rest-client
[4] https://github.com/dhuszti/SentimentAnalysisHUN

```
{
  "results": [
    {
      "input sentence": "A bajnok csapatból Stephen Curryt választották az elmúlt év
legjobb játékosának. Az ellenfél legjobb játékosa LeBron James sérülése miatt sajnos
nem játszhatott a döntőben.",
      "negative probability": 0.10785739058451087,
      "positive probalitiy": 0.89214260941548962,
      "sentiment": "positive"
    },
    {
      "entity": "Stephen Curryt",
      "entity type": "person",
      "negative prob": 0.15265205159468487,
      "positive prob": 0.84734794840531602,
      "sentiment": "positive"
    },
    {
      "entity": "LeBron James",
      "entity type": "person",
      "negative prob": 0.88357112376108005,
      "positive prob": 0.11642887623891852,
      "sentiment": "negative"
    }
  ]
}
```


5. ábra. Példa az alkalmazás részletes kimenetére

2. Haris Papageorgiou Suresh Manandhar Ion Androutsopoulos Maria Pontiki, Dimitrios Galanis. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 486–495, Denver, Colorado, 2015. Association for Computational Linguistics.
3. Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeny Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, pages 19–30, San Diego, California, June 2016. Association for Computational Linguistics.
4. Josef Steinberger, Tomáš Brychcın, and Michal Konkol. Aspect-level sentiment analysis in czech. *ACL 2014*, page 24, 2014.
5. Meishan Zhang, Yue Zhang, and Duy-Tin Vo. Neural networks for open domain targeted sentiment. In *Proceedings of the 2015 Conference on EMNLP*, pages 612–621, 2015.
6. Duy-Tin Vo and Yue Zhang. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 1347–1353, 2015.
7. Váradi Tamás Miháltz Márton. Trendminder: politikai témájú facebook üzenetek feldolgozása és szociálpszichológiai elemzése. In *XI. Magyar Számítógépes Nyelvi Konferencia*, MSZNY 2015, pages 195–198, Szeged, Magyarország, Januar 2015. Szegedi Tudományegyetem.
8. Fülöp Éva Kővágó Pál Miháltz Márton Váradi Tamás Pólya Tibor, Csertő István. A véleményváltozás azonosítása politikai témájú közösségi médiában megjelenő szövegekben. In *XI. Magyar Számítógépes Nyelvi Konferencia*, MSZNY 2015, pages 198–209, Szeged, Magyarország, Januar 2015. Szegedi Tudományegyetem.
9. Berend Gábor Hangya Viktor, Farkas Richárd. Entitásorientált véleménydetekció webes híranyagokból. In *XI. Magyar Számítógépes Nyelvi Konferencia*, MSZNY 2015, pages 227–234, Szeged, Magyarország, Januar 2015. Szegedi Tudományegyetem.
10. Miháltz Márton. Opinhubank: szabadon hozzáférhető annotált korpusz magyar nyelvű véleményelemzéshez. In *IX. Magyar Számítógépes Nyelvi Konferencia*,

MSZNY 2013, pages 343–345, Szeged, Magyarország, Januar 2013. Szegedi Tudományegyetem.

11. Yanqing Chen and Steven Skiena. Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 383–389, 2014.
12. Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. Polyglot-NER: Massive multilingual named entity recognition. *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, British Columbia, Canada, April 30 - May 2, 2015*, April 2015.
13. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
14. Peter Halacsy, Andras Kornai, and Csaba Oravecz. HunPos: an open source trigram tagger. In John Carroll and Eva Hajicova, editors, *Proc. ACL 2007 Demo and Poster Sessions*, pages 209–212. ACL, Prague, 2007.
15. Viktor Trón, György Gyepesi, Péter Halácsky, András Kornai, László Németh, and Dániel Varga. Hunmorph: Open source word analysis. In *Proceedings of the ACL Workshop on Software*, pages 77–85. Association for Computational Linguistics, Ann Arbor, Michigan, 2005.