Magyar nyelvű tulajdonnév-felismerés maximum entrópia módszerrel

Varga Dániel[1], Simon Eszter[2]

[1] Budapesti Műszaki Egyetem -- Média Oktató és Kutató Központ
daniel@mokk.bme.hu
[2] BME Kognitív Tudományi Tanszék
esimon@cogsci.bme.hu

Kivonat: Cikkünkben bemutatunk egy maximum entrópia módszeren alapuló statisztikai tulajdonnév-felismerő rendszert magyar nyelvre. A rendszer bemenetként morfológiailag elemzett szöveget dolgoz fel, ráépülve a hunpos morfológiai egyértelműsítőre. A felismerés hatásfokát tulajdonnév-címkézett magyar nyelvű korpuszon értékeljük.

1 Bevezetés

A tulajdonnév-felismerés (named entity recognition, NER) a természetes nyelv feldolgozását célzó alkalmazások közül az egyik legnépszerűbb, mivel hatékonyan automatizálható, és eredménye hasznos bemenete különböző magasabb szintű információ-kivonatoló és információ-feldolgozó rendszereknek.

A NER során egy bemeneti tokensorozatban kell tulajdonnevet alkotó intervallumokat (chunk) kijelölünk, ezeket véges sok kategóriába besorolva (például személy, szervezet, hely, egyéb). Egy NER algoritmus kiértékelése manuálisan annotált korpusszal való összevetés útján történik, és szokásosan maga az algoritmus is egy ilyen korpuszból tanulja meg a paramétereit automatikus módon.

Angol nyelvre sok tucat cikket publikáltak NER eljárásokról. Szinte minden ma ismert gépi tanulási módszert felhasználtak már NER címkéző építéséhez. Néhány példát említve: Rejtett Markov modellt használtak a BBN IdentiFinder építői [7], valamint Zhou és Su [3]; maximum entrópia módszert Borthwick [1] [2] és Chieu és Ng [3]; döntési fát Sekine et al. [8] [9]. Ezeknek a rendszereknek kevés a nyelvfüggő elemük, elvileg könnyen adaptálhatóak új nyelvekre. Magyar nyelvre ennek ellenére egyetlen kvantitatív vizsgálatról van tudomásunk: Szarvas et al. [11] publikált eredményeket Boosting és C4.5 döntési fa módszerekre épülő tulajdonnév-felismerő technológiájukról. Az általuk konstruált rendszer state-of-the-art pontosságot ér el angol nyelvre, és magyar nyelvre adaptálva a Szeged NER korpuszon [10] 94.77% CoNLL F-mértékű a pontossága. (A mérték definícióját lásd [13].)

Az alábbiakban felvázoljuk a rendszerünk architektúráját, leírjuk az általunk használt jegyeket, majd a mérési metodológia bemutatása után publikáljuk méréseink eredményeit.

2 A korpusz

Statisztikai alapú tanulórendszerünk tanításához és kiértékeléséhez a Szeged NER korpuszt [10] alkalmaztuk, cikkünk leadásának pillanatában ez volt az egyetlen gépi tanításhoz megfelelő méretű magyar nyelvű tulajdonnév-korpusz.[14]

A Szeged NER korpusz a Szeged Korpusznak [4] egy több mint 220 ezer szövegszónyi tematikusan válogatott és manuálisan tulajdonnév-annotált része. A szöveg legjellemzőbb tulajdonsága tematikus homogenitása: kizárólag gazdasági hírek szerepelnek benne. Ennek következtében nagyon nagy számban találhatók benne intézménynevek (magyar és multinacionális vállalatok, hivatalos szervezetek stb.), ez a kategória gyakoriságban dominálja a többit.

A Szeged NER Korpusz a CoNLL [13] típusrendszerét és címkézési konvencióit követi. Ennek megfelelően a következő kategóriákkal dolgoztunk: személynevek (PER), intézménynevek (ORG), földrajzi nevek (LOC) és egyéb tulajdonnevek (MISC), utóbbiak leggyakrabban márkanevek, címek és tőzsdeindexek megnevezései.

3 Tulajdonnévtárak

Rendszerünk fejlesztéséhez a különböző forrásokból származó tulajdonnévtárakat (gazetteer, a továbbiakban szótár) gyűjtöttünk össze:

- vezetéknevek
- keresztnevek
- magyar településnevek
- magyar nyelvű országnevek
- magyar utcanevek
- magyar cégnevek
- nemzetközi cégnevek
- cégnévvégződések (Rt., Kft., Ltd.)
- utcanévvégződések (út, utca, tér)
- pénzügyi rövidítések

Az első hat esetben forrásunk egy magyar telefonkönyv aggregált változata, illetve egy webes adatbázis volt. A cégnevek és utcanevek listáját automatikus eszközökkel megtisztítottuk a végződésektől, amelyek külön végződéslistába kerültek. A nemzetközi cégneveket tartalmazó listát Szarvas György és munkatársai bocsátották rendelkezésünkre.

Az egyetlen eset, amikor a fejlesztő (devel) korpuszon elkövetett hibák elemzése új szótár felvételéhez vezetett, a pénzügyi rövidítések tára. A devel korpuszban rendkívül gyakran szereplő tőzsdeindex-nevek (DAX, Libor, Nasdaq) MISC helyett sokszor tévesen ORG-nak címkézte az algoritmus. A „felületes szemlélő“ ezeket jó okkal minősítheti formájuk és szövegkörnyezetük alapján ORG-nak. Mint látni fogjuk, rendszerünk nem használ közvetlen módon a tanítókorpuszból épített szótárat,

[14] Reményeink szerint ez a helyzet hamarosan változni fog a HunNER korpusz megépítésével.

így ezeknek az egyedi eseteknek a megtanulása nehézséget okozhat a számára. A probléma megoldására egy weben található tőzsdei rövidítésgyűjteményben automatikus módon azonosítottuk a tanítókorpuszban ORG-ként soha nem szereplő elemeket, és ezekből szótárat alkottunk. Megjegyezzük, hogy a szótár alkalmazása nem javította, sőt kis mértékben rontotta a tesztelési korpuszon elért teljesítményt.

A rendszerünk végső, itt ismertetett változatában található szótárak (a fenti esettől eltekintve) teljesen azonosak a rendszer fejlesztésének megkezdése előtt véglegesítettekkel. Ennek oka a következő: bár a fejlesztés során komoly túl- és alulgenerálásokat találtunk a szótárakban, és ezeket sokszor javítottuk is, de azt tapasztaltuk, hogy az így elért pontosságnövekedés nem számottevő, és módszertani előnyei okán inkább visszatértünk a korpuszra nem rátanult szótárakhoz.

Az általunk épített szótárakat a rendszer forráskódjához hasonlóan szabad forráskódú licensz alatt publikáljuk.

4 Architektúra

Rendszerünk az [1] vagy [3] által ismertetett rendszerek architektúráját követi. Ha statisztikus gépi tanulási implementációt tervezünk, el kell döntenünk, hogy egy címkézési döntés meghozatalához milyen információt használjunk fel. Ez a feladat a jegykinyerés (feature extraction). NER esetében kézenfekvő és standard megoldás, hogy a vizsgált token környezetében levő tokenekről nyerünk ki nagy mennyiségű hasznosnak remélt egyszerű információt, például hogy nagy kezdőbetűs-e, szerepel-e valamely szótárunkban vagy mi a szófaja. Ezután valamilyen gépi tanulási algoritmusra bízzuk, hogy a tanulókorpusz alapján kiválogassa, hogy a nagy mennyiségű összegyűjtött jegyből melyek és hogyan segítenek a címkézési döntésben.

Gépi tanulási algoritmusként a maximum entrópia módszert választottuk. Ennek kimenete valószínűségeloszlás a választható címkéken, amelyből egy úgynevezett simító eljárással választjuk ki a legmegfelelőbbet.

4.1 Jegykinyerés

Megközelítésünk az volt, hogy minél nagyobb méretű, de egyszerűen implementálható jegyhalmazt építünk, kihasználva, hogy a maximum entrópia módszer nagyon nagy számú jegy hatékony feldolgozására képes. Jegyeink nagy része elsősorban a szavak egyszerű formai tulajdonságait írja le. Ugyanakkor azt is kihasználtuk, hogy rendelkezésünkre áll a hunpos morfológiai egyértelműsítő [6], amely ismeretlen szavak elemzését is elvégzi.

Az alábbi információkat építettük bele az algoritmusunkba:

1. Előfordul-e a token környezete valamely tulajdonnévtárunk valamely tételében, és ha igen, akkor a token a kifejezésnek mely pozícióján szerepel: az elején, a végén, a belsejében vagy egyszavasként? A többszavas kifejezések utolsó szavával való illeszkedést nem betű szerinti egyezésként definiáltuk, hanem egy alkalmasan választott kezdőszeleten való egyezésként.

2. Mondat eleje, mondat vége. (A Szeged Korpusz mondatra szegmentált, és a korpusz más felhasználóihoz hasonlóan ezt az információt az algoritmus számára hozzáférhetőnek tekintettük.)
3. A token igen/nem értékű formai jegyei: nagybetűs, csupa nagybetűs, tartalmaz kisbetű-nagybetű szekvenciát (pl. iPod), szám, számmal kezdődik, számot tartalmaz, kötőjelet tartalmaz, pontot tartalmaz.
4. A token felszíni, karakterlánc értékű formai jegyei: a token karakterszáma, a szóalak, a token három és öt hosszú kezdőszelete, a szó összes három karakterből álló összefüggő részkarakterlánca.
5. A hunpos morfológiai egyértelműsítő által szolgáltatott információk: szófaji kategória (NOUN, ART, NUM, ADJ, VERB, stb.), a szónak a hunpos egyértelműsítő által javasolt lemmája. Felismerte-e a hunmorph morfológiai elemző a szóalakot? Az azonosított lemma más kapitalizációjú-e, mint a token maga?

Az aktuálisan vizsgált token tehát megkapja a fent leírt jegyek közül azokat, amelyeket saját tulajdonságai implikálnak. Ezen kívül megkapja azokat a jegyeket is, amelyek a környezetében, de a mondatán belül álló szavak jegyei, mellékelve azt az információt, hogy mekkora eltolásra lévő szóból származik a jegy (pl. *oov.pre4*, *allcaps.post3*, *multi.start.pre1*). A figyelembe vett tokenek ablakának méretét optimalizáltuk; kényelmi okokból csak két paraméterre: egyrészt a karakterlánc értékű formai jegyek összességére, másrészt az összes többi jegyre. Méréseink szerint a karakterlánc értékű formai jegyeknél a 3 sugarú környezet (7 token) jegyeinek figyelembe vétele vezet optimális eredményhez, a többi jegy esetében az optimális az 5 sugarú környezet (11 token). Természetesen ezek az értékek nem univerzálisak, de modellünk többféle paraméterbeállítása mellett is a legkedvezőbbnek bizonyultak.

4.2 Címkék

A NER mint címkézési feladat eredeti formájában kevésbé alkalmas alanya gépi klasszifikálási módszereknek, mint az alábbi átcímkézett változatában: Minden tokent az alábbi 17 osztály egyikébe kell sorolnunk: {0, LOC.egytagú, LOC.eleje, LOC.belseje, LOC.vége, ORG.egytagú, …, MISC.vége}. Ennek a megoldásnak két előnye van a nyers ötelemű címkézéshez képest. Egyrészt a tanulóalgoritmus számára könnyebb felismerni olyan korrelációkat, amelyek speciálisan a tulajdonnév elejére és végére jellemzőek. Másrészt ez a címkézés implicit konzisztenciafeltételeket tartalmaz: például *.belseje után nem következhet *.eleje. Mint látni fogjuk, ezt felhasználhatjuk arra, hogy a gépi tanuló algoritmus kimenetét utófeldolgozva javítsuk.

4.3 Maximum entrópia

Címkézési algoritmusnak a maximum entrópia módszert választottuk. Ez a módszer már jó eredményeket hozott a hunpos morfológiai egyértelműsítő rendszer súlyozott morfológiai elemző (WMA) komponensének megépítésekor. Az általunk választott implementáció ezúttal könnyű beépíthetősége és nagy sebessége okán Zhang Le [15] rendszere volt, amely tanításhoz az L-BFGS algoritmust [17] alkalmazza.

Adathalmazainkon a L-BFGS iteratív tanulóalgoritmus 100 alatti iterációszámnál még nem kezd el konvergálni, a modell teljesítménye a pontos iterációszámtól erősen és kiszámíthatalanul függ. Publikált számaink 300-as iterációszám mellettiek, itt már tipikusan stabilizálódnak a modellek, és azok teljesítménye. Ugyanakkor a 300 iteráció relatíve magas, közel egy órás futásideje miatt az egyes elemi változtatások hasznos mivoltát gyakran kis (30 vagy 100) iterációszámmal épült modelleken vizsgáltuk, ami esetenként téves döntésekhez is vezethetett.

A futásidő kapcsán említjük meg, hogy a kezdőszeletek és karakter-trigramok jegyként való felvétele, és a nagyméretű ablakok alkalmazása miatt a jegyek száma rendkívül magas. A 200,000 példát tartalmazó tanítási korpuszon 250,000 különböző jegy összesen 10 millió előfordulása szerepel.

4.4 Simítás

A maximum entrópia klasszifikáló alkalmas arra, hogy több, helyességi valószínűséggel súlyozott alternatív javaslatot tegyen a címkére. Ennek fontos előnye, hogy erre épülve úgynevezett simítást implementáltunk, felülbírálva olyan lokális döntéseket, amelyek egymással inkonzisztensek. A gépi tanulási szakirodalomban elterjedt [3] módszer lényege, hogy 0 valószínűségű eseménynek tekintjük a lehetetlen átmeneteket (pl. LOC.belseje után ORG.eleje következik), uniform eloszlásúnak tekintjük a valid átmeneteket, és a maximum entrópia módszer által az egyes tokenekre kibocsátott valószínűségeloszlásokat függetlennek tekintve Viterbi módszerrel kiszámoljuk, hogy a mondatra mi a legnagyobb valószínűségű címkézéssorozat. Ez szükségszerűen valid lesz. Méréseink szerint ez a paramétermentes utófeldolgozási lépés egy tipikus mérési konfigurációban 0.5% körüli értékkel javítja meg rendszerünk F-pontszámát.

5 Mérések

5.1 Módszertan

A tanulóalgoritmus hatékonyságának méréséhez egy tesztkorpuszt kell címkéznie az algoritmusnak. A korpuszban rendelkezésre álló aranymérték (gold standard) címkék alapján mérhető az algoritmus pontossága és fedése. A NER rendszerek hatásfokának mérésére hagyományosan alkalmazott CoNLL kiértékelési függvény egy az algoritmus által azonosított tulajdonnevet címkéjének helyessége és a lefedett intervallum pontos megtalálása alapján is pontoz (0, ½ vagy 1 ponttal). Ezen pontszámok összesítése alapján egyesített pontossági és fedési értéket állapít meg, és az ezekből kapott F-pontszám a hatékonyság végső mérőszáma.

A Szeged NER korpusz birtokában egy ad hoc módon választott train-test szétválasztással, és keresztértékeléssel kezdtük meg rendszerünk vizsgálatát és fejlesztését. Később azonban világossá vált, hogy ha eredményeinket össze kívánjuk mérni az egyetlen rendelkezésre álló alternatívával, akkor a Szarvas et al. [11] által használt bontást és train-devel-test metodológiát kell használnunk.

Szarvas György és kollégái a rendelkezésünkre bocsátották az általuk alkalmazott adatszétválasztást, és ettől a ponttól ezt az övékkel azonos metodológiát alkalmaztuk, sőt a teljes összehasonlíthatóság céljából az általuk választott train-devel szétválasztást is átvettük. Minthogy erőforrásainkat a fejlesztés kezdetekor véglegesítettük, és a metodológia-váltás a paraméter-hangolás korai szakaszában történt, ezért meggyőződéssel állítjuk, hogy rendszerünk nem „fertőződött" meg a tesztadatok ismeretével.

A fejlesztés folyamán a tesztadathalmazt és az azon vétett hibákat nem tekintettük meg. A rendszer nagyszámú paraméterének optimalizálásakor szigorúan a devel halmazon elért pontosság (F-pontszám) maximalizálása vezérelt.

5.2 Eredmények

Ismertetett rendszerünk F-pontszáma 96.35% a devel korpuszon, és 95.06% a test korpuszon. Ezek kis mértékben magasabb értékek a [11] által publikált 96.20% illetve 94.77% F-pontszámoknál. Megjegyezzük azonban, hogy a [11] rendszer optimalizálása angol és magyar nyelvre párhuzamosan történt, ezzel szemben mi kizárólag a Szeged NER Korpusz adatain dolgoztunk, és rendszerünk nyilvánvalóan továbbfejlesztést igényelne ahhoz, hogy más nyelvű adatokon is jó eredményt adjon. Ezek a továbbfejlesztések terveink között szerepelnek.

1. Táblázat.

NE-típus	Devel	Test	Szarvas et al. Devel	Szarvas et al. Test
LOC	92.06	96.36		95.07
MISC	93.58	85.12		85.96
ORG	97.62	96.20		95.84
PER	97.44	94.94		94.67
Össz.	**96.35**	**95.06**	**96.20**	**94.77**

6 Köszönetnyilvánítás

Köszönetet mondunk Szarvas Györgynek és Farkas Richárdnak adathalmazaik megosztásáért, és Halácsy Péternek a simítási algoritmus kivitelezéséért.

Bibliográfia

1. A. Borthwick.: A Maximum Entropy Approach to Named Entity Recognition. PhD thesis, New York University, 1999.
2. A. Borthwick, J. Sterling, E. Agichtein, R. Grishman: NYU: Description of the MENE Named Entity System as Used in MUC-7. Proceedings of MUC-7, 1998
3. Hai Leong Chieu, Hwee Tou Ng: Named Entity Recognition with a Maximum Entropy Approach. In: Proceedings of CoNLL-2003, Edmonton, Canada, 2003, pp. 160-163.

4. Dóra Csendes, János Csirik and Tibor Gyimóthy: The Szeged Corpus: A POS tagged and Syntactically Annotated Hungarian Natural Language Corpus. Proceedings of TSD 2004, vol. 3206 (2004) 41-49.
5. P. Halácsy, A. Kornai, Cs. Oravecz, V. Trón, D. Varga: Using a morphological analyzer in high precision POS tagging of Hungarian. Proceedings of LREC 2006, pp. 2245—2248, 2006.
6. P. Halácsy, A. Kornai, D. Varga: Morfológiai egyértelműsítés maximum entrópia módszerrel Proc. 3rd Hungarian Computational Linguistics Conf., 2005. Szegedi Tudományegyetem.
7. S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone, R. Weischedel, and the Annotation Group (BBN Technologies): BBN: Description of the SIFT System as Used for MUC-7. Proceedings of MUC-7, 1998.
8. S. Sekine: Description of the Japanese NE System Used for MET-2. Proceedings of MUC-7, 1998.
9. S. Sekine, R. Grishman, and H. Shinou. A decision tree method for finding and classifying names in japanese texts. In Proceedings of the Sixth Workshop on Very Large Corpora, Montreal, Canada, 1998.
10. Gy. Szarvas, R. Farkas, L. Felföldi, A Kocsor, János Csirik: A highly accurate Named Entity corpus for Hungarian. Proceedings of International Conference on Language Resources and Evaluation, 2006.
11. Gy. Szarvas, R. Farkas and A. Kocsor: A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In Proceedings of Discovery Science 2006, DS2006, LNAI 4265 pp. 267-278, Springer-Verlag 2006.
12. E. F. Tjong Kim Sang, and Fien De Meulder: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, Proceedings of CoNLL-2003, 2003. 13. http://www.cnts.ua.ac.be/conll2003/ner/
14. Trón, Gy. Gyepesi, P. Halácsy, A. Kornai, L. Németh, D. Varga: Hunmorph: open source word analysis. Proceeding of the ACL 2005 Workshop on Software, 2005.
15. Zhang Le.: Maximum Entropy Modeling Toolkit for Python and C++. http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html
16. G. Dong Zhou, Jian Su: Named Entity Recognition using an HMM-based Chunk Tagger. Proceedings of the 40th Annual Meeting of the ACL, Philadelphia, pp. 473-480, July 2000
17. C. Zhu, R. H. Byrd, P. Lu, J. Nocedal: Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software (TOMS). 1997.