Hol a határ?
Mondatok, szavak, klinikák

Orosz György, Prószéky Gábor

MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport,
Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar
1083, Budapest Práter utca 50/a.
e-mail:{oroszgy, proszeky}@itk.ppke.hu

Kivonat Napjainkban egyre több elektronikusan rögzített dokumentum keletkezik klinikai környezetben, melyek egyik közös jellemzője, hogy létrehozásuk során a klinikai dolgozók nem fordítottak figyelmet a dokumentumok struktúrájának kialakítására, illetve a helyesírási normák betartására. Bár a mondat- és szóhatárok megállapítása egy olyan alapvető feladat, mely a feldolgozási lánc legelején helyezkedik el, irodalma mégsem jelentős, mivel ezt gyakran mérnöki munkának tekintik a kutatók. Jelen írásunkban ismertetjük a klinikai dokumentumok sajátosságait, különös tekintettel a mondat- és szóhatárok kérdésére. Részletesen bemutatunk egy hibrid szegmentáló algoritmust, mely szabályalapú részek mellett nem felügyelt gépi tanulást is használ. Az ismertetett módszer eredményességét részletesen megvizsgáljuk, másrészről összevetjük azt más magyar nyelvre elérhető rendszerekkel. Megmutatjuk, hogy a komplex eljárás teljesítménye jelentős mértékben meghaladja az alapjaként szolgáló szabályalapú rendszerét. Összevetve más mondatszegmentáló (és tokenizáló) metódusokkal, megállapítjuk, hogy csak az ismertetett új algoritmus képes oly mértékben mondat- és tokenhatárok azonosítására, hogy az a gyakorlatban is használható legyen.

1. Bevezetés

Magyarországon a napról napra keletkező nagy mennyiségű klinikai dokumentumok jelentős hányada csak archiválási célból készül és nem kerül feldolgozásra. Ezek nyelvtechnológiával támogatott újrafelhasználása, más nyelvekhez hasonlóan, nagy mértékben képes lenne segíteni a klinikákon praktizáló orvosokat jobb diagnózisok vagy új terápiák kifejlesztésében. A feldolgozó- és információkinyerő-eljárások legtöbbje a bemeneti szöveget mondatokra és/vagy szavakra bontva várja, így ezek pontos elvégzése szükségszerű. Bár az általános nyelvre léteznek nagy teljesítményű szegmentáló eszközök, de ezek alkalmazhatósága klinikai szövegeken nem bizonyított.

Írásunkban megvizsgáljuk a klinikai környezetben készült rekordokat, rávilágítva azok különleges tulajdonságaira. Bemutatunk egy kis méretű korpuszt, melyet az eszközök fejlesztése céljából hoztunk létre, majd ismertetünk

egy nagy teljesítményű szegmentáló algoritmust. Az eljárás szabályalapú komponenseken túl gépi tanuló (GT) algoritmusokat is foglalkoztat. Az utóbbi módszer alapja, hogy a nyers szövegekben pontra végződő tokenekről meghatározza, hogy a pont és a szó egybeírása csak a véletlen műve (mondathatár) vagy pedig szisztematikus használat eredménye (rövidítés). A pontosabb és teljesebb feldolgozás érdekében az eljárás számos más jellemző mellett morfológiai elemzéseket is használ.

A tesztkorpuszon végzett kiértékelésünkben megmutatjuk, hogy a klinikai szövegeken egyetlen szabadon elérhető eszköz sem teljesít megfelelően, míg az általunk fejlesztett algoritmus a gyakorlatban is jól használható.

2. Kapcsolódó munkák

2.1. Mondatok és tokenek azonosítása

A szövegek alkotóelemeinek keresése két részfeladatból tevődik össze: mondathatárok azonosítása és tokenekre bontás. Nagyon gyakran egy mondathatárkereső algoritmus feltételezi a rövidítések ismeretét, vagy magában foglalja azok azonosítását is. Míg a tokenizálást gyakran mérnöki feladatként kezeljük, ezzel szemben a mondathatárok felismerésének bővebb irodalma van. Read et al. összefoglaló írásában [1] az alábbi csoportokba osztja az ezzel foglalkozó kutatások: 1) szabályalapú rendszerek, amik domén- vagy nyelvspecifikus tudást használnak; 2) felügyelt gépi tanuláson (FGT) alapuló algoritmusok; 3) felügyelet nélküli gépi tanulást (FNGT) használó módszerek.

A gépi tanulást (GT) alkalmazó megoldások közül az egyik első Riley [2] algoritmusa volt, melyben döntési fákat használt mondatvégi írásjelek osztályozására. Analóg megközelítéssel bír a SATZ [3] keretrendszer, melyben számos FGT módszer érhető el, ami ezeken túl a szófaji címkék mint jellemzők használatára is képes. Az első eredmények, melyek maxent tanulást használtak mondatok szegmentálására, Reynar és Ratnaparkhi nevéhez fűződnek [4]. Másrészről a Gillick által bemutatott algoritmus [5] hasonló jellemzőket használva SVM módszeren alapul. Ismeretesek még Mikheev munkái, melyek közt szerepel egy szabályalapú eszköz [6], illetve ennek integrált használata egy szófaji egyértelműsítő keretrendszerben [7]. Az általunk ismert egyetlen FNGT-on alapú módszert Kiss és Strunk készítette, mely többszavas kifejezéseket azonosító algoritmust használ annak eldöntésére, hogy egy szó és egy pont rövidítést alkot-e.

Magyarra az ezidáig publikált alkalmazások szabályalapú megközelítést használnak: a huntoken [8] eszköz Mikheev rendszerén [6] alapul, míg a `magyarlanc` [9] hasonló modulja a MorphAdorner projekt [10] eredményeire épít.

2.2. Orvosi szövegek feldolgozása

Magyar nyelvű orvosi szövegek feldolgozásának irodalma ezidáig nem jelentős: Siklósi et al. [11,12] megoldása automatikus módon képes klinikai szövegek helyesírásának javítására, míg Orosz et al. egy morfológiai egyértelműsítő

rendszer teljesítményének növeléséről számolnak be [13]. Orvosi szövegek automatikus szegmentálásának kérdését egyik mű sem érinti.

Magyartól eltérően, az angol nyelvű orvosi szövegek szegmentálásának irodalma bővebb: mondatra bontó eljárásokként leginkább szabályalapú (pl. [14]) vagy FGT-t használó módszereket [15,16,17,18,19] használnak. Ezek közül is a legnépszerűbbek a maximum entrópián és CRF-en alapulók. A felügyelt tanuló algoritmusok egyik előnytelen tulajdonsága, hogy nagy mennyiségű manuálisan annotált adatra van szükségük. Ezek közül a doménspecifikus tanító anyagot használók általában jobban teljesítenek, de egyes kutatók, mint Tomanek et al. [20] az általános nyelvi adatok használata mellett érvelnek.

3. Erőforrások és metrikák

Az elkészült módszer fejlesztése és kiértékelése céljából szükséges volt létrehozni egy megfelelő méretű etalon korpuszt, illetve meghatározni azokat a metrikákat, amik a kiértékelés alapját képezték. Ebben a fejezetben ismertetjük az etalon létrejöttének lépéseit, jellemző tulajdonságait, majd pedig bemutatjuk azon mértékeket, melyek a méréseink alapját képezték.

3.1. Az etalon korpusz

A korpusz egy szemészeti klinikai rekordjainak véletlenszerűen kiválasztott bekezdéseit tartalmazza, melyeket először automatikusan tokenekre és mondatokra bontottunk, majd az így kapott szövegeket manuálisan javítottuk és ellenőriztük. Az így kapott etalon a helyesen szegmentált bekezdéseken túl tartalmazza még azok eredeti formáját is. A tesztkorpusz mintegy 2300 mondatot tartalmaz, melyből 1200 az egyes algoritmusok kiértékeléséhez, míg a maradék azok optimalizálására került felhasználásra.

Mivel az orvosi rekordokból kinyert bekezdések zajosak, így azok szegmentálása előtt szükség volt egy normalizáló modul alkalmazására is. Ennek a szabályalapú komponensnek az alábbi hibákkal kellett megküzdenie:

1. duplán konvertált karakterek, mint pl. '>',
2. „írógépproblémák": az '1' és '0' gyakran 'l' és 'o' betűkként szerepeltek,
3. dátumok nem konvencionális használata pl. '2011.01.02.', vagy '06.07.12.',
4. központozási hibák pl. '1.23mg', Törőközegek.Fundus :ép.'.

Hogy teljesebb képet kapjunk az orvosi szövegek karakterisztikájáról, összevetettük az etalont a Szeged Korpusszal (SZK) [21]. Az összehasonlítás az alábbi jelentős különbségeket fedte föl:

1. A rövidítések aránya az általunk vizsgált klinikai szövegekben mintegy 2,68%, míg ez az általános nyelvi korpuszban kevesebb mint 0,01% volt.
2. A SZK mondatai szinte mindig (98,96%) mondatzáró írásjellel végződnek, míg ez az orvosi szövegek mondataiban csak az esetek 51,72%-ban igaz.

3. Hasonlóan az előzőekhez, a mondatkezdő nagybetűk használatának aránya is nagymértékű eltérést mutat: a klinikai rekordokban ez csupán 87,19% míg az általános nyelvi szövegekben 99,58%.
4. A tokenizálást érintő jelentős különbség még a numerikus adatokat tartalmazó mondatok aránya, mely a klinikai rekordokban 13,50%, míg a SZK esetében ez az arány elhanyagolható.

3.2. Kiértékelési módszerek

A szakirodalomban nincs egyetértés afelől, hogy milyen metrikát érdemes használni a mondatrabontás és tokenizálás feladataiban: a GT módszereket alkalmazók gyakran F-mértéket, pontosságot és fedést használnak, míg beszédfelismerési feladatok esetén ugyanerre pl. a NIST metrikát alkalmazzák. Sokszor a fedés, illetve pontosság használata esetén sem egyértelmű, hogy mik az osztályozandó entitások, és azok milyen kategóriákba kerülhetnek.

Írásunkban a Read et al. [1] által bemutatott módszernek egy módosított változatát használjuk. Így a szegmentálást egy egységes osztályozási problémaként értelmezzük, amiben minden karaktert, illetve a köztük lévő üres sztringeket egy-egy címkével illetünk aszerint, hogy az entitás két token határán áll-e, egy mondatot zár-e le vagy esetleg az előzőek egyike sem. Ezt a sémát használva az eredmények elemzéséhez a bevett fedés- és pontosságalapú mértékekre támaszkodunk. A kiértékelés során az F_β-értéket is kalkulálunk: míg a tokenizálás feladatában az általános F_1 vizsgálatát megfelelőnek találtuk, a mondatokra bontás esetén a pontosságot előnyben részesítve a $\beta = 0,5$-t találtuk optimálisnak. Az utóbbi döntés mögött az a megfontolás áll, hogy a nyelvtechnológiai feldolgozási lánc rákövetkező moduljai még képesek lehetnek két szét nem választott mondat helyes elemzésére, de fals mondattöredékek feldolgozása a hibák további keletkezését szolgálja.

4. A szegmentáló lánc

Ebben a fejezetben ismertetjük azt az összetett algoritmust, mely nagy pontossággal végzi a klinikai szövegek mondatokra bontását. Az alábbiakban bemutatott algoritmus első eleme egy olyan szabályalapú komponens, ami elsősorban a tokenizálásért felelős. Ennek leírása után ismertetjük még azokat módszereket is, melyek tovább növelik a szegmentáló lánc teljesítményét.

4.1. A baseline algoritmus

Eljárásunk első lépésként egy olyan szabályalapú modult használ, melynek célja, hogy tokenekre bontsa a bekezdések szövegeit. A komponens ezen működését itt nem részletezzük, mivel algoritmusa tokenizálási feladatokban jól ismert szabályokra támaszkodik. Ez a komponens a tokenizáláson túl magában foglalja még olyan mondatvégek felismerését is, melyekre a tokenhatárok megállapítása során lehetőség nyílik. Erre a következő esetekben van mód:

1. ha egy létrejött token mondatvégi írásjel, ami egy nem írásjelet tartalmazó token előtt szerepel,
2. vagy ha egy sor egy teljes dátumkifejezéssel vagy egy vizsgálati eredménnyel kezdődik.

Megvizsgálva a fenti eljárás eredményességét azt találtuk, hogy így a mondatvégek mindössze felét lehetséges felfedni, ami az algoritmus magas pontossága mellett is túl alacsony összesített teljesítmény. A hibák részletes elemzése megmutatta még, hogy a fel nem ismert tokenhatárok jelentős része egybeesik a nem azonosított mondathatárokkal, ami szükségessé teszi a pontra végződő tokenek osztályozását. Így tehát úgy döntöttünk, hogy egy olyan komponenssel egészítjük ki az algoritmust, mely képes megkülönböztetni a rövidítéseket a mondatvégi szavaktól.

4.2. Eredményesebb mondathatár-felismerés gépi tanulás használatával

Általános nyelvi szövegekben kétfajta indikátor létezik, amik mondathatárokat jelezhetnek. Ez egyik ilyen az írásjelek jelenléte, a másik pedig a nagybetűk használata. Esetünkben az írásjelek közül csak a pont igényel további vizsgálódást, hiszen ez esetben áll csak fenn többértelműség. Hasonlóan a kapitalizált szavak elemzésével is körültekintően kell eljárni, hiszen a tulajdonneveken kívül az orvosi szövegekben bizonyos rövidítések és latin szavak is tévesen nagy kezdőbetűvel vannak írva. A fentieken felül nehezítik még a feladatot az olyan mondathatárok, amiknél mindkét jellemző egyszerre hiányzik.

Az indikátorokra építve is lehet automatikus eljárásokat építeni anélkül, hogy doménspecifikus rövidítéslista vagy tulajdonnév-szótár a rendelkezésünkre állna. Ugyanis egy feldolgozó algoritmusnak elégséges megfelelő bizonyítékot találnia egy szó (w), és az őt követő pont ($\bullet$) szeparáltságára, ami pedig Kiss és Strunk algoritmusához [22] vezet. Így tehát a kollokációk azonosítására használt log-likelihood arány egy megfelelő módszer a feladat megközelítésére. Esetünkben ez a (3)-ban formalizálható, ami statisztikai tesztre épülve felhasznál egy null és egy alternatív hipotézist.

$$H_0 : P(\bullet|w) = p = P(\bullet|\neg w) \tag{1}$$

$$H_A : P(\bullet|w) = p_1 \neq p_2 = P(\bullet|\neg w) \tag{2}$$

$$log\lambda = -2log\frac{L(H_0)}{L(H_A)} \tag{3}$$

A (1) formula a *(szó, $\bullet$)* pár függetlenségét fejezi ki, míg (2) teljesülése esetén feltételezhetjük, hogy ezek együttállása nem csupán véletlenszerű, mivel rövidítést jelölnek. Kiss és Strunk kutatása megmutatta, hogy a (3)-ban számolt $log\lambda$ értékek eloszlása χ^2-tel aszimptotikus, így statisztikai tesztként is használható. Ezzel együtt azt is megállapították, hogy ennek a módszernek a pontossága önmagában alacsony, így szükséges további skálázó faktorok alkalmazása.

Kutatásunkban ezekre az eredményekre támaszkodva alkalmazzuk a $log\lambda$ kalkulust, viszont szemben az eredeti munkával egy inverz pontozási módszert használunk ($iscore = 1/log\lambda$). Tesszük ezt azért, mert nem célunk az összes orvosi rövidítés azonosítása, sőt éppen ellenkezőleg, csak azon párok fellelése, amikről nagy biztonsággal feltételezhetjük, hogy nem összetartozóak, így tehát nem rövidített szóalakok. A fejlesztés során szükségesnek találtuk még a skálázó faktorok adaptálását is, melyet az alábbiakban részletezünk.

Hasonlóan [22]-hoz, az első tényező a tokenek hosszára épülve (len) jutalmazza a rövideket és bünteti a hosszúakat. A faktor számítása során felhasználtuk még a korpusz általános jellemzőit: az optimalizációs adatokból kinyert és manuálisan ellenőrzött rövidítéslista elemeinek a 90%-a legfeljebb 3 hosszúságú, míg az ettől hosszabb rövidített tokenek csak elvétve fordulnak elő. Így formalizáltuk ezeket a megfigyeléseket a (4) tényezőben.

$$S_{length}(iscore) = iscore \cdot \exp\left(len/3 - 1\right) \tag{4}$$

Mint azt [13]-ben ismertettük, a HuMor tőtárát orvosi doménen használatos szavakkal bővítettük, így ennek elemzéseit is felhasználtuk az osztályozási feladatban. Mivel az elemző számos rövidítést is ismer, így erre a tudásra alapozva tovább szűrhetjük a mondatvégi tokenek listáját. Az (5) indikátorfüggvény a HuMor elemzései alapján jelez, hogy az adott szónak létezik-e rövidítésre visszavezethető felbontása. A lexikális tudás nagyobb biztonsági foka miatt, nagyobb súlyt társítottunk ehhez a faktorhoz, továbbá (6) úgy került kialakításra, hogy képes legyen ellensúlyozni a rövid mondatvégi szavak hibás osztályozását.

$$indicator_{morph}(w) = \begin{cases} 1 & \text{ha } w \text{ szó elemzései között nincsen rövidítés} \\ -1 & \text{ha } w\text{-nek van rövidítés elemzése} \\ 0 & \text{egyébként} \end{cases} \tag{5}$$

$$S_{morph}(iscore) = iscore \cdot \exp\left(indicator_{morph} \cdot len^2\right) \tag{6}$$

A harmadik és egyben utolsó tényező a kötőjelek használatára épül. Vizsgálataink során azt tapasztaltuk, hogy ezek jelenléte nem jellemző a rövidítésekben, viszont annál inkább előfordulhatnak az összetett szavak képzésekor. Ezt a megfigyelést formalizálva a szó hosszával arányos tényezőként készítettük (7)-et, melyben a $indicator_{hyphen}$ akkor és csak akkor vesz fel 1 értéket, ha a szó tartalmaz kötőjelet, egyéb esetben az értéke 0.

$$S_{hyphen}(iscore) = iscore \cdot \exp\left(indicator_{hyphen} \cdot len\right) \tag{7}$$

A fentiek módosítók használatával számoljuk az összesített pontozást, amit (8) mutat be. Az $sscore$-t minden ponttal végződő tokenre kalkulálja az algoritmus, majd összeveti ezt egy empirikusan meghatározott küszöbértékkel ($< 1,5$), mely alapján rövidítésnek azonosítható egy entitás.

$$sscore = S_{hyphen} \circ S_{morph} \circ S_{length}(iscore) \tag{8}$$

4.3. További kapitalizáción alapuló szabályok

Munkánkban létrehoztunk még egy olyan komponenst is, mely szavak kapitalizációjára támaszkodik. Ez a modul is épít a HuMorra: ha egy szó analízisei között nem szerepel egy tulajdonnévi elemzés sem, és a szó nagy kezdőbetűvel van írva, akkor a szóban forgó entitás mondatkezdő jelöltté válik. Ezek további szűrésére is szükség van, mivel fennáll még a veszélye annak, hogy egy több tagból álló tulajdonnév egyik elemével van dolgunk. Így a kontextusok figyelembevételével, csak azokat a szavak kerülnek a mondatkezdő osztályba, amik biztosan nem tulajdonnevek.

5. Eredmények

Az algoritmus egészének teljesítményére egy mutató az összesített pontosság. Az 1. táblázatban közreadjuk az előfeldolgozott és a szegmentáló metódusok eredményeinek megfelelő értékeit. Itt a pontosság értékek magas volta azzal magyarázható, hogy a kiértékelő módszer a leggyakoribb jelenséget (*nincs módosítás*) egyformán jutalmazza a legnehezebbekkel. Közelebbi képet kapunk a komponensek egyenkénti teljesítményéről a 2. táblázatban, amiben a hibarátájuk csökkenését prezentáljuk.

1. táblázat. Az egyes feldolgozási fázisok összesített pontossága

	Összesített pontosság
Előfeldolgozott adat	97,55%
Baseline algoritmus	99,11%
Teljes lánc	99,74%

2. táblázat. Az egyes rendszerek hibaarányának csökkenése a baselinehoz viszonyítva

	Hibaráta csökkenés
$(w, \bullet)$ párok osztályozásával	58,62%
Kapitalizáción alapuló szabályokkal	9,25%
A teljes lánc	65,50%

Tüzetesebben megvizsgálva az egyes modulok teljesítményét a hagyományos pontosság, fedés és F-értékeket is számolunk. A mondathatárok azonosítását tekintve a 3. táblázat értékei jelentős teljesítménynövekedésről számolnak a fedést illetően, míg pontossági értékek csak kis mértékben csökkennek.

Eredményeinket érdemes tanulmányozni más magyar nyelvre szabadon elérhető szegmentáló eszközök teljesítményének fényében is. Vizsgálatunkban a

3. táblázat. Az egyes mondatrabontó modulok eredményességének vizsgálata

	Pontosság (P)	Fedés (R)	$F_{0,5}$
Baseline	96,57%	50,26%	81,54%
$(w, \bullet)$ párok osztályozásával	95,19%	78,19%	91,22%
Kapitalizáción alapuló szabályokkal	94,60%	71,56%	88,88%
A teljes lánc	93,28%	86,73%	91,89%

teszthalmaz adatain kiértékeltük a `magyarlanc` megfelelő modulját, a huntoken eszközt, az OpenNLP[1] mondatrabontó komponensét, illetve Punkt nyelvfüggetlen rendszert. A huntoken rendszer a működéséhez rövidítéslistákat használ, mely lehetőséget adott működésének testreszabásához. Így vizsgálatunk kiterjedt az általános tokenizáló (HTG) teljesítményén túl, egy orvosi rövidítésekkel adaptált (HTM) verziójára is. Mivel az OpenNLP FGT algoritmusokat használ mondatvégek azonosítására, így ehhez tanítóanyagként a Szeged Korpuszt mondatait használtuk.

4. táblázat. Szabadon elérhető mondatrabontó alkalmazások teljesítményének kiértékelése

	Pontosság (P)	Fedés (R)	$F_{0,5}$
`magyarlanc`	72,59%	77,68%	73,55%
HTG	44,73%	49,23%	45,56%
HTM	43,19%	42,09%	42,97%
Punkt	58,78%	45,66%	55,59%
OpenNLP	52,10%	96,30%	57,37%
A hibrid lánc	93,28%	86,73%	91,89%

A 4. táblázat adatai azt sugallják, hogy a zajos orvosi szövegeken az általános nyelvhasználatra optimalizált szoftverek sikertelennek bizonyulnak. Bár az OpenNLP kiemelkedő fedéssel rendelkezik, de cserébe a mondatok majd felét hibásan vágja szét, ami végeredményben alacsony F-pontot eredményez. Robusztus teljesítményt mutat még a `magyarlanc`, mely eredmény a jól felépített, doménfüggetlen szabályok használatának köszönhető. Ezekkel szemben a huntoken egyes változatai nyújtják a legalacsonyabb pontosságot és F-pontokat is. A Punkt eredményeit vizsgálva azt találjuk, hogy a felügyelet nélküli tanuló algoritmus doménadaptációja mintegy kétszeres teljesítménynövekedést eredményezett.

Bár munkánkban főleg a mondatok szegmentálására koncentrálunk, de vizsgáltuk még a tokenizáló rendszerek pontosságát is. Az elvégzett mérések (5. táblázat) összhangban állnak azzal a feltételezésünkkel, hogy a baseline algoritmus által fel nem fedezett tokenhatárok jelentős része egyben mondathatár is.

[1] http://opennlp.apache.org/

5. táblázat. A tokenizálás feladatára vonatkozó eredmények

	Pontosság (P)	Fedés (R)	F_1
Baseline	99,74%	74,94%	85,58%
A teljes lánc	98,54%	95,32%	96,90%

6. Összegzés

Írásunkban ismertettünk egy hibrid algoritmust, mely kiemelkedő eredményességgel képes mondat- és tokenhatárok azonosítására klinikai rekordok bekezdéseiben. Vizsgálatunk célja elsősorban a mondatvégek helyes detektálása volt, melyhez egy három lépésből álló eljárást készítettünk. A készített feldolgozási lánc szabályalapú komponensek mellett felügyelet nélküli gépi tanulásra is támaszkodik. Az algoritmus első lépésben mintaillesztés használatával elvégzi az alapszintű tokenizálást, majd ennek eredményében az egyes *(szó, •)* párok eloszlását figyelembe véve azonosítja a mondathatárok nagy részét, melyet az utolsó szabályalapú komponens tovább finomít. A bemutatott algoritmus különlegessége, hogy a határkeresési feladatokhoz egy morfológiai elemző tudását is sikerrel használja.

A létrehozott rendszer teljesítménye, összehasonlítva más szabadon elérhető szoftverekkel szemben is, kiemelkedően magas. Vizsgálatunk megmutatta, hogy a létrejött hibrid algoritmuson kívül nincsen más olyan szabadon hozzáférhető eszköz, mely hasonló eredményességgel végezné orvosi szövegeken a szegmentálás feladatát.

Köszönetnyilvánítás

Ez a munka részben a TÁMOP – 4.2.1.B – 11/2/KMR-2011-0002 és TÁMOP – 4.2.2/B – 10/1–2010–0014 pályázatok támogatásával készült.

Hivatkozások

1. Read, J., Dridan, R., Oepen, S., Solberg, L.J.: Sentence Boundary Detection: A Long Solved Problem? In: 24th International Conference on Computational Linguistics (Coling 2012). India. (2012)
2. Riley, M.D.: Some applications of tree-based modelling to speech and language. In: Proceedings of the Workshop on Speech and Natural Language, Association for Computational Linguistics (1989) 339–352
3. Palmer, D.D., Hearst, M.A.: Adaptive sentence boundary disambiguation. In: Proceedings of the fourth conference on Applied natural language processing, Association for Computational Linguistics (1994) 78–83
4. Reynar, J.C., Ratnaparkhi, A.: A maximum entropy approach to identifying sentence boundaries. In: Proceedings of the fifth conference on Applied natural language processing, Association for Computational Linguistics (1997) 16–19

5. Gillick, D.: Sentence boundary detection and the problem with the US. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, Association for Computational Linguistics (2009) 241–244
6. Mikheev, A.: Periods, capitalized words, etc. Computational Linguistics **28**(3) (2002) 289–318
7. Mikheev, A.: Tagging sentence boundaries. In: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference, Association for Computational Linguistics (2000) 264–271
8. Halácsy, P., Kornai, A., Németh, L., Rung, A., Szakadát, I., Trón, V.: Creating open language resources for Hungarian. In: Proceedings of Language Resources and Evaluation Conference. (2004)
9. Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: Proceedings of Recent Advances in Natural Language Processing 2013, Hissar, Bulgaria, Association for Computational Linguistics (2013) 763–771
10. Kumar, A.: Monk project: Architecture overview. In: Proceedings of JCDL 2009 Workshop: Integrating Digital Library Content with Computational Tools and Services. (2009)
11. Siklósi, B., Orosz, Gy., Novák, A., Prószéky, G.: Automatic structuring and correction suggestion system for hungarian clinical records. In De Pauw, G., De Schryver, G.M., Forcada, M.L., M Tyers, F., Waiganjo Wagacha, P., eds.: 8th SaLTMiL Workshop on Creation and use of basic lexical resources for lessresourced languages. (2012) 29.–34.
12. Siklósi, B., Novák, A., Prószéky, G.: Context-aware correction of spelling errors in hungarian medical documents. In Dediu, A.H., Martín-Vide, C., Mitkov, R., Truthe, B., eds.: Statistical Language and Speech Processing. Volume 7978 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 248–259
13. Orosz, Gy., Novák, A., Prószéky, G.: Magyar nyelvű klinikai rekordok morfológiai egyértelműsítése. In: IX. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem (2013) 159–169
14. Xu, H., Stenner, S.P., Doan, S., Johnson, K.B., Waitman, L.R., Denny, J.C.: Medex: a medication information extraction system for clinical narratives. Journal of the American Medical Informatics Association **17**(1) (2010) 19–24
15. Apostolova, E., Channin, D.S., Demner-Fushman, D., Furst, J., Lytinen, S., Raicu, D.: Automatic segmentation of clinical texts. In: Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE, IEEE (2009) 5905–5908
16. Cho, P.S., Taira, R.K., Kangarloo, H.: Text boundary detection of medical reports. In: Proceedings of the AMIA Symposium, American Medical Informatics Association (2002) 998
17. Savova, G.K., Masanz, J.J., Ogren, P.V., Zheng, J., Sohn, S., Schuler, K.K., Chute, C.G.: Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. Journal of the American Medical Informatics Association **17**(5) (2010) 507–513
18. Taira, R.K., Soderland, S.G., Jakobovits, R.M.: Automatic structuring of radiology free-text reports. Radiographics **21**(1) (2001) 237–245
19. Tomanek, K., Wermter, J., Hahn, U.: Sentence and token splitting based on conditional random fields. In: Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics. (2007) 49–57

20. Tomanek, K., Wermter, J., Hahn, U.: A reappraisal of sentence and token splitting for life sciences documents. Studies in Health Technology and Informatics **129**(Pt 1) (2006) 524–528
21. Csendes, D., Csirik, J., Gyimóthy, T.: The Szeged Corpus: A POS tagged and syntactically annotated Hungarian natural language corpus. In: Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora. (2004) 19–23
22. Kiss, T., Strunk, J.: Unsupervised multilingual sentence boundary detection. Computational Linguistics **32**(4) (2006) 485–525