Normo: Egy automatikus normalizáló eszköz középmagyar szövegekhez

Vadász Noémi[1,2], Simon Eszter[1]

[1]MTA Nyelvtudományi Intézet
[2]MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport
E-mail: {vadasz.noemi, simon.eszter}@nytud.mta.hu

Kivonat A cikk egy automatikus normalizáló eszközt ismertet középmagyar szövegek normalizálásához. A Normo két modulból áll: egy memóriaalapú modulból és egy szabályalapúból, amely karakter- és tokenszintű környezetfüggő újraíró szabályokat tartalmaz. Az eszköz meggyorsítja és megkönnyíti a középmagyar szövegek kézi normalizálását, amelynek eredménye a további nyelvfeldolgozó eszközök bemenete. Az eszköz ismertetése után a modulok teljesítményét külön-külön és egyben is kiértékeljük.
Kulcsszavak: normalizálás, szabályalapú normalizálás, memóriaalapú normalizálás, történeti szövegek, középmagyar

1. Bevezetés

Az annotált nyelvi erőforrások elérhetősége egyre fontosabb szerepet kap a nyelvészet több területén: a nyelvtechnológiai fejlesztéseken kívül az elméleti és történeti nyelvészeti kutatásoknak is kiváló alapanyagot szolgáltatnak a korpuszok. A történeti korpuszok az adatok és a nyelvi jelenségek gazdag tárházát adják – de csak akkor, ha a releváns információ elektronikusan interpretálható és előhívható módon van tárolva bennük. A nyelvtörténészek és nyelvtechnológusok egyik legfontosabb együttműködési terepe a történeti korpuszok építése. Az elmúlt évtizedekben sorra indultak olyan projektek, melyek egy adott nyelv valamely régebbi változatának digitalizálását és feldolgozását célozták – elsősorban indoeurópai nyelvekre, például [1,2]. Ebbe a sorba illeszkedik az Ómagyar Korpusz [3] is, amely tartalmazza az összes fennmaradt ómagyar kori (896–1526) szövegemléket, valamint bizonyos középmagyar kori (1526–1772) írott és nyomtatott szövegeket, továbbá néhány szövegemlék normalizált és morfológiailag elemzett és egyértelműsített változatát[1].

Napjainkban a korpuszépítési munkálatok során elsősorban már digitalizált szövegekből indulnak ki; de nem ez a helyzet a történeti dokumentumokkal. Az elektronikus formátumok (sőt az elektromosság) előtti korból származó szövegekkel való foglalkozás sokkal idő- és munkaigényesebb folyamat, és bizonyos

[1] http://omagyarkorpusz.nytud.hu/

esetekben más módszereket is igényel, mint a mai szövegek esetében. A helyesírás és a központozás a régebbi nyelvváltozatok korában nem volt konzisztens, ezért a modern szövegekben alkalmazott sztenderd előfeldolgozó lépések (tokenizálás, mondatra bontás, morfológiai elemzés és egyértelműsítés) nem végezhetők teljesen automatikusan, és nagyon sok kézi ellenőrzést igényelnek.

A magyar írásosságot a latin nyelvű és vallásos tárgyú irodalom fordításának igénye hívta életre, de a latin ábécé magyarra alkalmazása számos problémát vetett fel. A legfőbb gond abból fakadt, hogy nyelvünk hangrendszerének több eleme a latinban ismeretlen, így ezek jelölésére új jeleket kellett bevezetni. A helyesírás ezekben a századokban távolról sem volt egységes. A különböző helyesírási rendszerekben is ritka az egy hang–egy betű megfelelés (vagyis amikor egy hang jelölésére mindig ugyanaz a betű használatos, és az adott betűnek mindig egy hangértéke van), de egy alakulóban levő helyesírási rendszerben ilyenfajta következetesség még annyira sem várható el. Ezért szükség van egy ún. normalizálási lépésre, amelynek során az eredeti betűhű szóalakokat mai magyar helyesírású szavakra alakítjuk át. A többféle, különböző nyelvtörténeti szakmai érvekkel alátámasztható lehetséges feldolgozási forgatókönyvek egyik gyakori közös átalakító lépése ez a fajta normalizálás [4,5]. A szövegfeldolgozásnak ez a lépése kritikus fontosságú, enélkül ugyanis a (félig) automatikus annotáció hatékonysága a következő lépésekben drámaian visszaesik [6].

Mivel a normalizálás nyelvtörténeti szakértelmet kívánó, rendkívül időigényes manuális munka, megpróbáltuk kiváltani gépi eljárással. A normalizálás nyelvtechnológiai szempontú kutatásának igen gazdag eszköztára van, mivel a normalizálási feladatot jellemzően más nyelvtechnológiai feladatokkal szokták párhuzamba állítani és azok eszközkészletét használni a feladat megoldására. Az egyik megközelítés szerint a normalizálás során egy forrásnyelvről (jelen esetben a régi nyelvváltozatról) fordítunk egy célnyelvre (a modern nyelvváltozatra), így a gépi fordítás módszerei használhatók [7,8]. Egy másik megközelítés a normalizálást egy ábécéről egy másikra való átírásként, vagyis transzliterációként fogja fel, amire a Shannon-féle zajos csatorna modellt kiválóan lehet alkalmazni [9,10]. A normalizálási munkák nagy része viszont kézzel írott vagy korpuszból kigyűjtött megfeleltetési szabályok [11] és/vagy távolságmetrikák [12] alkalmazásán alapul.

A cikkben egy olyan normalizáló eszközt ismertetünk, amelyet középmagyar szövegek kézi normalizálásának a támogatására fejlesztettünk. Jelen állapotában az eszköz a Károli Gáspár-féle középmagyar bibliafordításra lett optimalizálva, ami a teljes Biblia első magyar nyelvű fordítása 1590-ből. Ennek ellenére a NORMO nem csak egy memória- és szabályalapú eszköz ennek az egy szövegnek a normalizálásra, hanem egy olyan keret, amely további szótárak és szabályok hozzáadásával alkalmassá tehető más történeti szövegek normalizálására is. Nem kétséges, hogy a feladat a fent felsorolt technikák bármelyikével is megoldható volna, de mi különféle, elsősorban a projektből fakadó, praktikus kényszerek és követelmények hatására a memória- és szabályalapú technikák ötvözését választottuk.

A cikk az alábbiak szerint épül fel. A 2. fejezetben azt ismertetjük, hogy melyek azok az alfeladatok, amelyeket a kézi normalizálást végző annotátorok elvé-

geznek, vagyis meghatározzuk, hogy mi az ideálisan elérendő cél. A 3. fejezet a Normo által alkalmazott módszerekről szól: a 3.1. fejezet írja le a memóriaalapú modult, míg a 3.2. fejezet a szabályalapú normalizálást ismerteti. A 4. fejezetben a Normo alapos kiértékelését adjuk, összehasonlítva egy távolságmetrikát alkalmazó másik, történeti szövegek normalizálására kifejlesztett eszközzel. A cikket az 5. fejezetben összefoglalás zárja.

2. A kézi normalizálás

A kézi normalizálás során az annotátorok olyan feladatokat is elvégeznek, amelyek kívül esnek a szűkebb értelemben vett normalizálás feladatkörén, de a szöveg további feldolgozását készítik elő. Így a tulajdonképpeni normalizálással párhuzamosan bekerülnek a megfelelő lókuszjelölők a szövegbe, megtörténik a (tag)mondatokra bontás és a kis/nagybetűsítés is. A tokenizálás javítása is megtörténik, ahol a mai magyar helyesírásnak megfelelően bizonyos szavakat fel kell bontani, míg másokat össze kell vonni – természetesen jelölve a változtatásokat. Ahol szükséges, ott az annotátorok megjegyzést vagy értelmezést is adhatnak az egyes szavakhoz, valamint ebben a fázisban történik az igekötők kódolása is a megfelelő igéhez vagy igenévhez. Ennek megfelelően a szöveg kézi normalizálásán egy tágabb értelemben vett előkészítő munkát értünk, amely magába foglalja a szűkebb értelemben vett normalizálás mellett a fent ismertetett egyéb előkészítő feladatokat is.

A gépi normalizálás során ezekből az alfeladatokból néhányat ki tudunk váltani automatikus eszközökkel. A tokenizálás annyiban tud automatikusan történni, amennyiben azt az egyes tokenek felszíni jegyei lehetővé teszik. Egyrészt a tapadó írásjeleket leválasztjuk a szavakról, másrészt a sorvégi kötőjellel jelölt elválasztott szóelemeket összevonjuk, és speciális karakterekkel jelöljük az elválasztás és összevonás tényét és helyét. Bár a ma használatos logikai–grammatikai írásjelezés kibontakozása csak a 17. században kezdődik, a mondatra bontásban mégis csak a korabeli központozásra támaszkodhatunk, így minden írásjel után beszúrunk egy tagmondathatár-jelölő üres sort. A szűkebb értelemben vett normalizálás során ugyanazokat az elveket követjük, mint amiket az annotátorok követnek a kézi normalizálás során. Egyrészt a ma nem létező összes szót, toldalékot, morfológiai konstrukciót meg kell tartani, vagyis morfémát nem toldhatunk be és nem hagyhatunk el. Másrészt viszont el kell hagynunk minden fonológiai és helyesírási esetlegességet, vagyis egységes, amennyire lehet, a mainak megfelelő helyesírásra kell törekedni.

Mivel a normalizált szöveg képezi a további szövegfeldolgozó lépések bemenetét, a cél a lehető legpontosabban normalizált szöveg, ezért a normalizálást annotátorok végzik kézzel. Ugyanakkor a kézi normalizálás időigényes és drága feladat. Az automatikus normalizáló eszköz segítségével le lehet rövidíteni a normalizáláshoz szükséges időt és költségeket. Az annotátorok visszajelzései alapján az automatikus normalizáló kimenetének ellenőrzése és javítása nem csak gyorsabb, hanem egyszerűbb is a betűhű szöveg kézzel történő normalizálásánál, amikor minden egyes szóalakot egyesével kell begépelni. A Normo kimenetének kézi

javítása és a teljesen kézi normalizálás közötti munkaráfordításbeli különbségről lásd a 4. fejezetet.

3. Módszerek

A Normo két fő modulból áll: egy memória- és egy szabályalapú modulból. A szabályalapú modul karakter- és tokenszintű környezetfüggő szabályokból áll. A karakterszintű szabályok a tokeneken belül, a tokenszintű szabályok a szóhatárokon keresztül operálnak, sőt ez utóbbiak között vannak olyanok is, amelyek tagmondathatárt is változtatnak.

A tokeneken belüli változtatások kétféleképpen történhetnek: a memóriaszótár vagy a karakterszintű újraíró szabályok segítségével – vagyis ez utóbbi két modul egymástól független. Mindkettő kimenete viszont bemenetét képezheti a tokenszintű szabályoknak, amely kimenete adja a Normo végső kimenetét.

3.1. Memóriaalapú normalizálás

A Normo memóriaalapú moduljának az alapja egy szótár, amely a leggyakoribb szóalakokat és azok kézzel normalizált változatát tartalmazza egy `tsv` fájlban. A memóriaalapú modul legnagyobb előnye a pontosság, hiszen a szótárban szereplő összes szóalak kézzel lett normalizálva. Az automatikus normalizálás során azok a szóalakok, amelyek a szótárban is szerepelnek, egyszerűen lecserélődnek a szótárban szereplő megfelelőjükre. A szótár alapján normalizált szóalakoknál további javításra nincs szükség, így a további normalizáló eljárások (a karakter- és tokenszintű szabályok) számára stopszóként viselkednek.

Mivel a memóriaszótár kézzel készült, a lehető legmagasabb pontosságot várjuk el. A fedés viszont a szótár méretétől függ. A jelen cikkben bemutatott eredményekhez a Károli-biblia négy evangéliumának és az Újtestamentum bevezetésének a leggyakoribb szavaiból készítettük el a szótárt. Az automatikus normalizálás kézzel ellenőrzött kimenete a későbbiekben tovább növelheti a szótár méretét – és így a fedést is. A kiértékeléskor a szótár 502 szóalakot tartalmazott.

A memóriaalapú normalizálás hátránya, hogy a normalizálandó szóalak és a szótárban szereplő szóalak között teljes karakterszintű egyezés szükséges. Vagyis ha egy betűhű szó egyszer nagybetűvel, egyszer kisbetűvel szerepel a szövegben, de a szótárban csak nagybetűvel van, akkor a kisbetűseket elveszítjük. Ezt megelőzendő, a memóriaalapú normalizálás előtt kisbetűsítjük a szöveget. A memóriaalapú normalizálás alkalmazása után csak azok a szavak lesznek nagybetűsek, amelyeknek a normalizált megfelelője nagybetűs a szótárban, tehát a tulajdonnevek.

3.2. Szabályalapú normalizálás

A Normo szabályalapú modulja kézzel írt újraíró szabályokat tartalmaz. A szabályok két forrásból származnak: egyrészt nyelvtörténeti kutatások eredményeképpen, másrészt korpuszalapú megfigyelések alapján fogalmaztuk meg őket. Az

újraíró szabályok a lecserélendő karaktersorozat szomszédos karakterei vagy a szóhatárok alapján működnek, tehát környezetfüggők. Illeszkedés esetén a környezetfüggő szabály alapján az eredeti karaktersorozatot lecseréljük a normalizált karaktersorozatra. Ennek megfelelően a szabályok alkalmazása által eredményezett szóalak nem feltétlenül egy valid magyar szó.

Karakterszintű szabályok. A karakterszintű újraíró szabályokat egy tokenen belül alkalmazzuk. A szabályok között vannak olyanok, amelyek a középmagyar és a mai magyar karakterkészlet különbségeit oldják fel úgy, hogy a mai magyar karakterkészlet részét nem képező karaktereket (pl. *Æ* és *ę*) a mai magyarban megtalálható megfelelőjükre fordítja át (*e* és *é*). Ennek megfelelően a normalizált alakok már nem tartalmaznak olyan karaktereket, amelyek nem képezik részét a mai magyar karakterkészletnek. Továbbá olyan szabályok is vannak, amelyek a mai magyar helyesírásban tükröződő fonotaktikai szabályszerűségeket fedik le. Például szóvégen csak hosszú *ó* vagy *ő* szerepelhet, a rövid párjuk nem, így a karakterszintű szabályok segítségével minden szóvégi *o* és *ö* karaktert rendre *ó* és *ő* karakterre cserélünk.

A karakterszintű szabályok esetén nincs szükség teljes karakteregyezésre egy szóalakban, hiszen ezek a szabályok a szóalakon belül operálnak. Ennek megfelelően egy tő különböző végződésű változataiban ugyanúgy végbemennek a karakterszintű szabályok (ellentétben a 3.1. fejezetben említett memóriaalapú normalizálással). Ugyanakkor a karakterek együttes előfordulásai különböző viselkedést mutatnak annak megfelelően, hogy egy morfémán belül vagy morfémahatáron vizsgáljuk őket. A magyar nyelv agglutináló jellege miatt ezzel a problémával gyakran kell szembesülnünk. Például a morfémán belüli intervokális környezetben szereplő *i*-t *j*-re kell fordítanunk, morfémahatáron viszont *i*-nek kellene maradnia. A Normo nem kezeli különbözőképpen a morfémán belüli és a morfémahatáron álló karaktercsoportokat, ezért ezekben az esetekben számba kell vennünk a szabály alkalmazásával létrejövő helyes és helytelen szóalakok számát. Ha a szabály több helyes szóalakot hoz létre, mint helytelent, akkor a szabályt alkalmazzuk, majd a helytelen alakok javítását a kézi annotátorokra bízzuk.

A 88 darab karakterszintű újraíró szabályt reguláris kifejezések formájában implementáltuk. Mivel egy szóalakon belül több újraíró szabályt is alkalmazhatunk, valamint egy újraíró szabály kimenete bemenete lehet egy másiknak, ezért a reguláris kifejezések sorrendje nagyon fontos. Például a [ttz → tsz] és a [tz → c] szabályok kimenete az alkalmazásuk sorrendjétől függ. Ezért minden egyes szabály megfogalmazásakor számba kellett vennünk a korábban megfogalmazott szabályok be- és kimeneteit.

Tokenszintű szabályok. A tokenszintű szabályokat a fent ismertetett memóriaalapú és a karakterszintű normalizáló modulok kimenetén alkamazzuk. Ezek a szabályok a szóhatárokon keresztül operálnak, így módosítják a normalizált szöveg tokenszámát. A 7 darab tokenszintű szabály közül egy szóalakra csak egyet

alkalmazunk, szemben a 3.2. fejezetben ismertetett karakterszintű szabályokkal, amelyek közül akárhány alkalmazható egy szóalakon belül.

A tokenizálást befolyásoló szabályok között megfogalmaztunk egészen általánosakat is (pl. a felsőfokot jelölő *-leg* prefix és az azt követő szó összevonása), de vannak közöttük egyedi, de gyakori eseteket kezelő szabályok is (a *szent* szóalak és a *lélek* tövű szó egybeírása). A NORMO tokenizálást befolyásoló szabályait az 1. táblázat foglalja össze.

összevonás	betűhű	normalizált
igekötő + ige	***le*** *megy* →	***le****megy*
mutató nm. + kérdő nm.	***az ki*** →	***aki***
személyes nm. + névutó	***én utánam*** →	***énutánam***
felsőfok + melléknév	***leg*** *jobb* →	***leg****jobb*
szent + lélek	***szent lélek*** →	***szentlélek***
szétszedés		
szó + *is*	*porát****is*** →	*porát* ***is***
szó + *nélkül*	*orvos****nélkül*** →	*orvos* ***nélkül***

1. táblázat. A szavak tokenizálását befolyásoló szabályok.

A karakterszintű szabályokhoz hasonlóan a tokenszintű szabályok alkalmazása sem mindig helyes mai magyar szót eredményez. A tokenszintű szabályok számára problémát okozhat a homográfia, például a *meg* szó esetében, amely lehet igekötő vagy kötőszó. Míg igekötőként az őt követő igével egybeírandó, kötőszóként külön tokennek kellene maradnia. A homográfiából fakadó hibák javítását a kézi annotátorokra bízzuk.

A fentieken kívül vannak olyan tokenszintű szabályok is, amelyek a mondatra bontást befolyásolják. Ilyen szabály például az, amelyik a *hogy* kötőszó és a vonatkozó névmások elé vesszőt és tagmondatot jelző üressort illeszt. Jelen cikkben csak a szószintű tokenizálást végző szabályok teljesítményét értékeljük ki.

4. Kiértékelés

A NORMO teljesítményének kiértékeléséhez a Károli-biblia négy evangéliumát és az Apostolok cselekedeteit használtuk. A szövegek kézzel normalizált változatát vettük gold standardnak, azzal vetettük össze az automatikus normalizáló kimenetét. A szövegek tokenszáma a betűhű változat alapján 114 580 token, a normalizált változat alapján 109 289 token.

A kiértékeléshez két mérőszámot használtunk. A *normalizálási pontosság* mutatja a gold standardnak megfelelően normalizált tokenek arányát, így a tulajdonképpeni szószintű normalizálás teljesítményét. Ez a mérőszám a NORMO

moduljai közül a memóriaalapú modul és a szabályalapú modul karakterszintű szabályainak a teljesítményét méri.

A normalizálási pontosság kiszámolásakor figyelembe kellett vennünk azokat a szavakat is, amelyeket a normalizálás során akár a kézi annotátor, akár a Normo tokenszintű szabályai szétválasztottak vagy összevontak. Az eredeti betűhű szöveggel összevetettük a kiértékeléskor gold standardként használt kézzel normalizált szöveget és a Normo kimenetét is, így a szétválasztott vagy összevont szavak normalizálási pontosságát is ki tudtuk számolni.

A karakterszintű szabályok egyik rendszerben sem tudják jól kezelni a tulajdonneveket, mivel azok nem feltétlenül követik az adott nyelv fonológiai és helyesírási szabályait. Mivel a szöveg, amivel dolgozunk, bibliafordítás, nagyon sok bibliai nevet találunk bennük. A kézi normalizálás során a különböző bibliafordításokban és bibliai históriákban említett tulajdonnevek is normalizálásra kerültek, vagyis az adott nevek különbözőképpen használt alakjai a Szent István Társulati bibliafordításnak[2] megfelelően egységes formára lettek hozva. A hibák nagy részét a nevek adják – kezelésük még nem megoldott a Normo-ban.

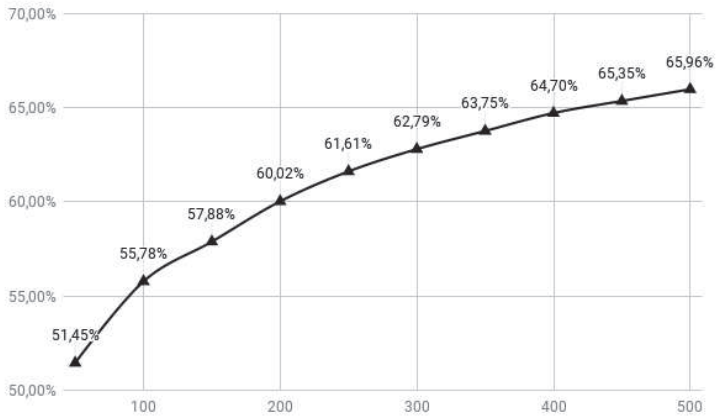
A szabályalapú modul tokenszintű szabályainak, tehát a tokenizálást befolyásoló szabályoknak a teljesítményét egy másik mérőszám fejezi ki. Ennek kiszámolásához három kategóriát állapítottunk meg: a) álpozitív: a Normo összevont két tokent, amelyek a gold standardban nincsenek összevonva, vagy szétválasztott egy tokent, amely a gold standardban nincs szétválasztva; b) álnegatív: a Normo nem vont össze két tokent, amelyek a gold standardban össze vannak vonva, vagy nem választott szét egy tokent, amely a gold standardban szét van választva; és c) valós pozitív: a Normo összevont két tokent, amelyek a gold standardban is össze vannak vonva, vagy szétválasztott két tokent, amely a gold standardban is szét van választva. Ezeknek a hibafajtáknak a számosságát ismerve kiszámoltuk a pontosságot, a fedést és az F-mértéket, amelynek eredményét a normalizálási pontossággal együtt a 2. táblázat mutatja.

	tokenszintű	tokenszint fölötti		
	pontosság (%)	pontosság (%)	fedés (%)	F-mérték (%)
Máté	81,04	87,90	69,40	77,56
Márk	80,62	86,18	65,80	74,62
Lukács	81,58	84,56	67,50	75,07
János	83,10	94,20	70,60	77,03
ApCsel	79,95	90,72	67,13	77,16
átlag	81,23	88,63	68,16	77,06

2. táblázat. A Normo kiértékelése a Károli-biblia egyes részein.

[2] http://szentiras.hu/SZIT

A kiértékeléskor a szótárunk 502 szóalakot tartalmazott. Feltételezzük, hogy minél nagyobb a memóriaalapú normalizáláshoz használt szótár, annál magasabb normalizálási pontosságot érhetünk el – ugyanakkor minimalizálni szerettük volna a szótár készítéséhez használt manuális munkát. Leteszteltük, hogy különböző méretű – ötvenesével növelt – szótárak használata mellett a NORMO memóriaalapú modulja önmagában milyen teljesítményt ér el, vagyis ennél a kiértékelésnél a szabályalapú modult nem működtettük. Az eredményeket az 1. ábra mutatja.



1. ábra: Az ötvenesével növelt szótárméret mellett kapott normalizálási pontosság a NORMO memóriaalapú moduljának kiértékelésekor. A vízszintes tengelyen a szótár mérete, a függőleges tengelyen a normalizálási pontosság látható.

Az eredmények alapján látható, hogy a NORMO 502 szóalakot tartalmazó szótára önmagában is elég magas normalizálási pontosságot ér el – ami még kiegészül a szabályalapú modulok teljesítményével is. Mivel a szótárba az egyes szóalakok gyakoriságuk alapján kerültek bele (tehát az 502 leggyakoribb szó), ezért a szótár további növelése nem okozna nagy javulást a normalizálási pontosságban – a szótárméret növelésével javuló normalizálási pontosság görbéje kilapulna.

A kimenetet emellett a NORMA automatikus normalizáló eszköz [12] eredményével is összevetettük. A NORMA is szabályalapú, de abban különbözik a NORMO-tól, hogy a karakterszintű újraíró szabályokat a tanítóanyagként megkapott, kézzel normalizált szövegből egy módosított Levenshtein-algoritmus segítségével tanulja ki. A tanítóanyag megegyezett a NORMO memóriaalapú moduljában használt szótárral. A NORMA a tanítóanyag mellett egy célnyelvi szólistát is használ, amely a Szent István Társulati bibliafordítás szavait tartalmazó szólista. A karakterszintű szabályok hatékonyságának összevetéséhez a NORMO esetében csak a szabályalapú modult működtettük, a memóriát nem használtuk. Az eredményeket a 3. táblázat tartalmazza.

	pontosság	
	Normo (%)	Norma (%)
Máté	68,29	61,58
Márk	66,43	61,70
Lukács	67,03	61,31
János	68,59	59,14
ApCsel	66,98	60,18
átlag	67,36	60,67

3. táblázat. A Normo és a Norma rendszerek normalizálási pontosságának összehasonlítása.

Az eredmények azt mutatják, hogy ekkora tanulóanyag rendelkezésre állása esetén a nyelvtörténeti tudás hozzáadása határozottan javít egy szabályalapú rendszer teljesítményén.

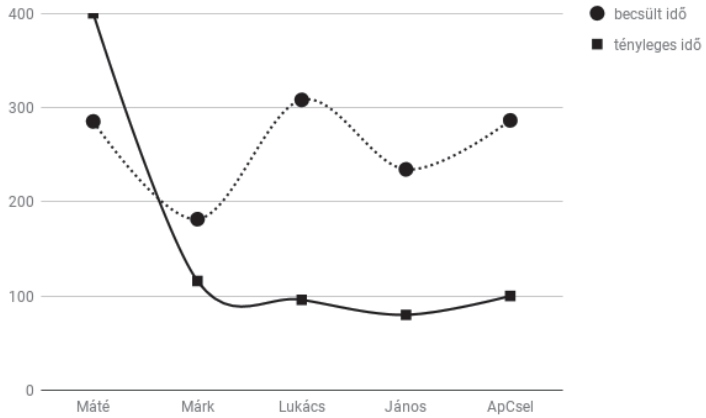
Egy olyan módszer kiértékelésének, amely a kézi munkát hivatott automatikusan segíteni, egy másik lehetséges dimenziója az, ha az adott feladatra szánt kézi munka mennyiségét és az automatikus módszerrel megtámogatott munka mennyiségét vetjük össze. Ezt azt összehasonlítást tartalmazza a 4. táblázat.

	tokenszám (#)	becsült idő (h)	tényleges idő (h)	arány (%)
Máté	28 520	285,2	400	140,25
Márk	18 150	181,5	116	63,91
Lukács	30 805	308,05	96	31,16
János	23 435	234,35	80	34,13
ApCsel	28 631	286,31	100	34,92

4. táblázat. A teljesen kézi normalizálás és a gépi normalizálás utáni kézi javítás időigényének összevetése.

A kiértékelés alapjául a betűhű szöveg tokenszámát vettük, mert mind a kézi, mind a gépi normalizálásnak az a bemenete. A becsült idő az Ómagyar Korpusz annotálásának eddigi tapasztalataiból leszűrt idő, ami 100 token óránként – ezt tekintjük annak az időtartamnak, ameddig akkor tartott volna a normalizálás, ha teljes mértékben kézzel zajlott volna. A tényleges idő azt mutatja, hogy mennyi ideig tartott a manuális javítási munka, amikor az annotátorok a Normo által normalizált szöveget kapták meg bemenetként. Az arányszámok pedig a tényleges és a becsült idő arányát mutatják. A számokból azt láthatjuk, hogy a Máté evangéliumának normalizálása igényelte a legtöbb időt, és az erre kapott arányszám a legnagyobb. Ez egyrészt annak köszönhető, hogy ennek a

résznek az automatikus normalizálása a Normo fejlesztésének egy korábbi szakaszában zajlott, így a kimenetben sokkal több kijavítandó hiba volt. Másrészt pedig annak, hogy a korábbi gyakorlattal ellentétben a Normo kimenetében minden szónak van normalizált megfelelője – ha nem történik a bemenő szón semmi változás, akkor önmaga –, és ez nehezebben kezelhető volt az annotátor számára, mintha nem lett volna ott semmi. A Normo fejlesztése során az egyre jobb minőségű kimenet egyre inkább megkönnyítette a kézi annotátorok dolgát, ami jól kirajzolódik a 2. ábrán látható diagramon.



2. ábra: A normalizáláshoz felhasznált idő változása az automatikus normalizálás bevezetésével. A vízszintes tengelyen a szövegek, a függőleges tengelyen a normalizálásuk ellenőrzéséhez felhasznált munkaidő látható órában kifejezve.

5. Összefoglalás

Cikkünkben egy automatikus normalizáló eszközt mutattunk be, amely egy memóriaalapú és egy szabályalapú modulból áll. A Normo-val normalizált és a kézi annotátorok által ellenőrzött középmagyar szöveg megfelelő bemenet lesz a további nyelvfeldolgozó eszközök számára. A Normo magas teljesítménye, valamint a gépi normalizálás utáni kézi javítás alacsony időigénye jól mutatja, hogy a Normo használata megkönnyíti és meggyorsítja a normalizálást.

A fejlesztés jelenlegi fázisában az automatikus normalizálót a Károli-bibliára adaptáltuk, de a szótár és a szabályok módosításával további középmagyar szövegekre is alkalmazható. Ezért a későbbiekben szeretnénk más szövegekre (például a többi középmagyar bibliára) is alkalmazni az eszközt, valamint újabb normalizálási eljárásokat is (pl. Levenshtein-alapú normalizálást, gépi tanulási megoldásokat) kipróbálni és kiértékelni.

Hivatkozások

1. Kroch, A., Taylor, A.: The Penn-Helsinki Parsed Corpus of Middle English (PPCME2) (2000) URL: http://www.ling.upenn.edu/ppche-release-2016/PPCME2-RELEASE-4.
2. Galves, C., Britto, H.: The Tycho Brahe Corpus of Historical Portuguese (2010) URL: http://www.tycho.iel.unicamp.br/ tycho/corpus/en/index.html.
3. Simon, E.: Corpus building from Old Hungarian codices. In É. Kiss, K., ed.: The Evolution of Functional Left Perpiheries in Hungarian Syntax. Oxford University Press (2014) 224–236
4. McEnery, T., Hardie, A.: Lancester Newsbooks Corpus. (2003) URL: http://www.lancs.ac.uk/fass/projects/newsbooks/default.htm.
5. Novák, A., Gugán, K., Varga, M., Dömötör, A.: Creation of an annotated corpus of Old and Middle Hungarian court records and private correspondence. Language Resources and Evaluation (2017)
6. Rayson, P., Archer, D., Baron, A., Culpeper, J., Smith, N.: Tagging the Bard: Evaluating the accuracy of a modern POS tagger on Early Modern English corpora. In: Proceedings of the Corpus Linguistics Conference (CL2007), UK (2007)
7. Bollmann, M., Søgaard, A.: Improving historical spelling normalization with bi-directional LSTMs and multi-task learning. In: Proceedings of COLING 2016: Technical Papers, Osaka, Japan (2016) 131–139
8. Pettersson, E.: Spelling Normalisation and Linguistic Analysis of Historical Text for Information Extraction. PhD thesis, Uppsala University, Department of Linguistics and Philology (2016)
9. Brill, E., Moore, R.C.: An improved error model for noisy channel spelling correction. In: Proceedings of ACL 2000. (2000)
10. Oravecz, Cs., Sass, B., Simon, E.: Semi-automatic Normalization of Old Hungarian Codices. In: Proceedings of the ECAI 2010 Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2010), Lisbon, Portugal (2010) 55–60
11. Bollmann, M., Petran, F., Dipper, S.: Rule-based normalization of historical texts. In: Proceedings of the Workshop on Language Technologies for Digital Humanities and Cultural Heritage, Hissar, Bulgaria (2011) 34–42
12. Bollmann, M.: (Semi-)Automatic Normalization of Historical Texts using Distance Measures and the Norma tool. In: Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2), Lisbon, Portugal (2012) 3–14