

Hibajavítási idő csökkentése magyar nyelvű diktálórendszerben

Szabó Lili¹, Tarján Balázs¹, Mihajlik Péter^{1,2}, Fegyő Tibor^{1,3}

¹ Budapesti Műszaki és Gazdaságtudományi Egyetem,
Távközlési és Médiainformatikai Tanszék,
{lili,tarjanb}@tmit.bme.hu

² THINKTech Kutatási Központ Nonprofit Kft.,
mihajlik@thinktech.hu

³ SpeechTex Kft.
tfegyo@speeche.com

Kivonat A gépi beszédfelismerésen alapuló diktálórendszerek természetes velejárója a felismerési hiba, melyet tipikusan a szófelismerési hibarárányal jellemzünk. A felhasználó számára azonban nem a klasszikus szóhibaarány a meghatározó mint használhatósági metrika, hanem sokkal inkább a hibajavítási idő. Cikkünkben azt vizsgáljuk, hogy valós, magyar nyelvű, relatíve kötött témájú (EU-s joganyagok) diktálási feladat esetén milyen faktorok befolyásolják elsődlegesen a hibajavítási időt, azt hogyan lehet csökkenteni. A saját rendszerünket összevetettük a piacon ingyenesen hozzáférhető magyar nyelvű diktálórendszerekkel. Megmutattuk, hogy a beszédfelismerési modellek feladatra szabásán túl az írásjelek, speciális rövidítések és egyéb szövegszerkesztési parancsok diktálhatóságának lehetővé tétele jelentősen csökkentheti a diktálásra fordított időt és energiát, így növelve a felhasználói elégedettséget.

1. Bevezetés

Cikkünk témája egy korábbi nagyszótáras, magyar nyelvre fejlesztett, folyamatos beszédfelismerőn alapuló e-mail diktálórendszer [1] továbbfejlesztése, valamint a diktálási feladat felhasználó számára történő megkönnyítése. A legfejlettebb technológiát alkalmazó, beszélőre adaptált, célfeladatra tanított automatikus beszédfelismerő rendszerek angol nyelvre, 90-95%-os felismerési pontossággal működnek. Egy, a beszédfelismerésen alapuló diktálást napi szinten, professzionális célokra használó felhasználó számára akár ennél magasabb felismerési pontosság is elégedetlenséghez vezethet, tekintve, hogy egy felismerési hiba észlelése és javítása akár 15-30 másodpercet is igénybe vehet [2]. Mivel a diktálási feladat természetes velejárója a felismerési hibák javítása, ezért a hibajavítás támogatása elengedhetetlen komponense egy diktálórendszernek. Jelen tanulmányunkban hibajavításon mind a szöveg utólagos formázását, mind a felismerő által ejtett hibák utólagos korrigálását értjük. Feltételezésünk szerint a felismerési kimenetben szereplő hibák észlelését nagymértékben könnyíti a szöveg jólformázottsága. Egy felismerési kimenetben alapértelmezésben nem szerepelnek sem

írásjelek, sem nagy kezdőbetűs alakok, illetve szövegszakaszokat határoló sortörések, és szükséges ezek helyreállítása ahhoz, hogy a felhasználó egy formázott szöveget tudjon létrehozni.

Természetesen már önmagában a felismerési hibák csökkentése is nagy szerepet játszhat a hibajavítási idő csökkentésében. Ennek, és a piaci termékekkel való összevethetőségnek az érdekében egy kötött témájú (*closed domain*), jogitörvénykezési diktálásra optimalizált rendszert építettünk, ami azért előnyös, mert a felismerési hiba csökkenését eredményezi, ezáltal lehetőséget teremtve a hibajavítási folyamatot támogató új módszerek kipróbálására. Magyar nyelvre ingyenesen hozzáférhető, beszédfelismerésen alapuló diktálórendszer a *Google* webalapú⁴, és a *Nuance* okostelefonra készített szövegbeviteli alkalmazásába⁵ integrált beszédfelismerési szolgáltatásaként érhető el. Tanulmányunkban ezeket vetjük össze rendszerünkkel, leginkább a hibajavítási folyamat szempontjából.

A 2. fejezetben beszédfelismerési kimenetek szerkesztésével foglalkozó legújabb kutatásokat tekintjük át. A 3. fejezet a diktálórendszerek kiértékelését végző metrikákat, és a SpeechTex rendszer felépítését írja le. A 4. fejezet a diktálórendszerek összehasonlításához végrehajtott kísérletek felépítését ismerteti. Az 5. és 6. fejezetek a kísérletek eredményeit és tanulmányunkban levont következtetéseket tartalmazzák.

2. Irodalmi áttekintés

A mondathatárok automatikus detektálása, nagy kezdőbetű- és írásjel-visszaállítás a beszédfelismerési kimenetben egy gyakran kutatott téma a szakirodalomban [3]. Gépi tanulási algoritmusokat használó módszerekkel ezen feladatokon elért pontosság 30-50% körül mozog [4,3]. A feladatot nehezítik a felismerési kimenetben különböző arányban előforduló hibák. Magyar nyelvre hasonló megoldást [5]-ben találhatunk. Ebben a kutatásban különböző modalitású tagmondat-típusokra HMM modelleket építettek, amelyek segítségével a tagmondatfajtákat felismerték. A felismeréshez felhasználták egy, a tagmondatok egymás utáni sorrendjét figyelembe vevő szöveg szintű prozódiai modellt is. 6 tagmondat-típus és egy szünetmodell, mellett 50%-os pontosságot értek el (úgy, hogy a helyesen felismert írásjelek aránya 70% körül mozgott). Az automatikus írásjelezésre alternatív megoldás, az írásjelek diktálhatóságának lehetővé tétele, melyet jelen fejlesztés során alkalmaztunk, igen magas pontossággal működik.

A hibajavítási folyamat támogatása ehhez szorosan kapcsolódó téma. Az eddigi kutatások azt tükrözik, hogy csupán a felismerési pontosság javítása nem elegendő, hiszen hibák mindig lesznek a felismerési kimenetben, ezért magát a hibajavítási folyamatot kell meggyorsítani és megkönnyíteni a felhasználó számára. Az egyik megközelítés a jelenség kezelésére a beszédfelismerő adaptálása a felhasználók javításait visszacsatolva: szótárban nem szereplő szavak hozzáadása a nyelvi modellhez, a nyelvi modell újraszűzítése, valamint kiejtési alternatívák

⁴ <https://www.google.com/intl/en/chrome/demos/speech.html>

⁵ <http://www.swype.com>

generálása [6]. A másik módszer a felismerési kimenet utógondozása. A hagyományos helyesírás-ellenőrzéstől abban lényegesen különbözik ez a feladat, hogy a beszédfelismerési kimenetben kizárólag olyan szavak fordulhatnak elő, amelyek szerepeltek a nyelvi modell tanításához használt korpuszban. Ebből következik, hogy a kimenetben előforduló hibák „valódi szavas” (*real word*) hibák, melyek kezelése egy, a kontextust is figyelembe vevő eljárást igényel. Számos módszer született már a probléma megoldására; a hagyományosnak tekinthető *noisy channel* [8] modellben egy mondat összes szavától adott szerkesztési távolságra lévő szavak potenciálisan helyes szavak, a javítás a legvalószínűbb szószorozat kiválasztásával történik, tetszőleges n -gram alapon.

Az automatikus beszédfelismerés felhasználó-központú, illetve a hibajavítási folyamat szempontjából történő kiértékelése egy aránylag kevés figyelmet kapó terület, [7] tartalmaz egy körültekintő tanulmányt különböző diktálási tapasztalattal rendelkező felhasználók újonnan elsajátított hibajavítási szokásaival, a hangsúly itt inkább az egyének közötti változatosságon van, mintsem a hibajavítási folyamat kvantitatív értékelésén.

3. Módszer

3.1. Kiértékelés

Szóhibaarány. A szóhibaarány (*word error rate - WER*) az automatikus beszédfelismerésen alapuló rendszerek egyik legnépszerűbb kiértékelési módszere. A szavak szintjén méri a hibás behelyettesítések (S), törlések (D) és beillesztések (I) számát a felismerési kimenetben, és ezek arányát a referenciában előforduló szavak számához (N) képest.

$$\text{Szóhibaarány} = \frac{S + D + I}{N} \quad (1)$$

Új metrikák. Egy diktálórendszer teljeskörű kiértékelése csak úgy lehetséges, ha az a felhasználó nézőpontját is figyelembe veszi. Ennek érdekében három új mérőszámot/metrikát vezettünk be, amik a hibajavítási folyamatot hivatottak kiértékelni:

1. **Szerkesztési Idő:** mennyi időt vesz igénybe a felhasználónak a felismerési kimenetben a hibákat megtalálni és javítani, valamint a szöveget jólformázott alakra hozni.
2. **Sikerességi Ráta:** milyen mértékben sikerül a felhasználónak a felismerési kimenetet a kívánt/eredeti szöveg alakjára hozni. A szóhibaarányhoz hasonlóan a behelyettesítések (S), törlések (D) és beillesztések (I) hibák karakterszintű számolása a már szerkesztett kimenetben (lényegében Levenshtein-távolság az eredeti szövegtől) elosztva/normalizálva az eredeti szövegben előforduló szavak számával (N), az írásjeleket is figyelembe véve.
3. **Gépelési Idő:** mennyi időt vesz igénybe ugyanazon eredeti szöveg gépelése másodpercben.

3.2. Rendszerek

A 1. táblázat összefoglalja a három diktálórendszer jellemzőit. Megjegyzendő, hogy míg a Google rendszer magyar nyelvre nem rendelkezik sem az írásjelek diktálhatóságának, sem az írásjelek automatikus helyreállításának funkciójával, a Nuance rendszer az automatikus helyreállítás jeleit mutatja, noha ez becslésünk szerint az esetek kevesebb, mint 10%-ban fordul elő.

1. táblázat. A három rendszer funkcióinak összehasonlítása.

Funkciók	Google	Nuance	SpeechTex
Írásjelek	–	automatikus diktálva	
Nagybetűsítés	–	automatikus diktálva	
Ütemezés	valós idejű	késleltetett	valós idejű
<i>Domain</i>	nyitott	nyitott	törvénykezés

3.3. Korpusz és normalizálás

A nyelvi modell építéséhez használt korpusz [9] egy többnyelvű adatbázis az európai parlamenti ülések leiratainak hivatalos fordításaiból, amiből a magyar ún. *fordítási egységeket* használtuk fel. A korpusz adatait a 2. táblázat foglalja össze.

2. táblázat. DGT-TM korpusz adatok

korpusz rész	normalizálás előtt		normalizálás után	
	token	type	token	type
tanító	35.3 M	1.3 M	43.3 M	645 K
dev	129 K	27 K	145 K	18 K
eval	94 K	21 K	114 K	15 K

A normalizálás első lépése a mondathatárok helyreállítása volt. Ez a mondatvégi pont és a rövidítések, valamint a mondatkezdő nagybetűs szó és a tulajdonnevek egymástól való elválasztásával történt, a korpuszban előforduló gyakoriságok alapján. Az ezt követő *tokenizálás* során a következő *token* típusokat különböztettük meg: szavak, tulajdonnevek, mozaikszavak, rövidítések, URL-ek, email-címek, számok, dátumok, jogi jelölések, speciális szimbólumok, egyéb nem nyelvi elemek. Ezek átalakítása szóveges alakra, valamint a beszélt formára nem alakítható egyéb nem nyelvi elemek eltávolítása reguláris kifejezések segítségével történt.

Duplikációk detektálása és eltávolítása a nyelvi modell simításához fontos, hogy megtörténjen, mert a simítási eljárás során használt *counts-of-counts*-ok

eloszlását zavarja, ha páros számú *count*-ok kiugróan magasabbak, mint a páratlanok, és a *count*-ok nem egyenletesen csökkenő eloszlást követnek.

3.4. Nyelvi modell

A nyelvi modell módosított Kneser-Ney simítás használatával készült az SRI Language Modeling Toolkit (SRILM) [11] segítségével. A létrehozott trigram (3-gram), szóalapú modellekben entrópiaalapú metszést egyetlen esetben sem alkalmaztuk.

3.5. Akusztikus modell

Az Egri Katolikus Rádió (EKR) beszélgetéseiből válogatott, összesen 43 óra hanganyagon tanított, környezetfüggő akusztikus modell a HTK [10] eszközeinek segítségével készült, ami összesen 6121 egyenként 13 Gauss-függvényből álló állapotot tartalmaz. A 16 kHz-en mintavételezett felvételek lényegkiemeléséhez 39 dimenziós, delta és delta-delta értékkel kiegészített mel-frekvenciás kepsztrális komponenseken alapuló jellemzővektorokat hoztunk létre, és vak csatornaki-egyenlítő eljárást is alkalmaztunk.

3.6. Hálózatépítés és dekódolás

A legvalószínűbb illeszkedés kereséséhez használt dekódolási folyamat szervertkliens-architektúra alapján működik. A rendszer által használt beszédfelismerő kliens (VOXclient) végzi a beszédalapú információ lényegkiemelését és a 3.5 fejezetben vázolt jellemzővektorokká való alakítását. A jellemzővektorokat ezután a kliens továbbküldi a szervertoldali alkalmazásnak (VOXerver), ahol a tényleges dekódolási lépések megtörténnek. A legvalószínűbb illeszkedés megtalálásához a beszédfelismerési modelleket ún. súlyozott, véges állapotú átalakítóknak (*Weighted Finite State Transducer* - WFST) [12] egyesítjük. A szervert a kliensoldal felé végül visszaküldi a megtalált legvalószínűbb felismerési kimenetet; minden frissítés 250 ms-onként zajlik. A normalizálás során átalakított nem verbális nyelvi elemek (számok, URL-ek) írott formára való visszaalakítása szintén a kliens oldalon történik a már visszaküldött legvalószínűbb felismerési kimeneten.

4. Kísérletek

A magyar nyelvre ingyenesen elérhető gépi beszédfelismerő rendszerek teljesítményének felhasználó szempontú összehasonlítását egy 6 résztvevős (3-3 férfi/nő, életkor: 22-38 év) kísérletben végeztük, amiben a résztvevők

1. egy rövid (7 mondatból álló) jogi szöveget olvastak fel 2 módban:
 - (a) normál olvasási mód és
 - (b) az írásjelek hangalakjának diktálásával, majd a hanganyagokat rögzítettük.

2. A felismerés
 - (a) a Google és Nuance rendszerek esetében a normál olvasási módban,
 - (b) SpeechTex rendszer esetében pedig az írásjelek hangalakjának diktálásával készült változatokon történt.
3. Ezeken a kimeneti szövegeken zajlott aztán a hibajavítási feladat, ami a felismerési hibák detektálását, javítását és egyéb szövegszerkesztési műveleteket foglalja magába.
4. A beszédfelismerésen alapuló diktálási tapasztalatokról végül egy kérdőívben kérdeztük a résztvevőket, amiben egy Likert-alapú skálán (az adott állítással való egyetértés erősségének kifejezése egy 1-5-ig terjedő intervallumban) kellett értékelniük a diktálási feladatot és a kísérletben szereplő diktálórendszerek teljesítményét.

5. Eredmények

Nem parametrikus páros, egymintás Mann-Whitney-Wilcoxon tesztekkel ellenőriztük, hogy a rendszerek közti szóhibaarányok szignifikánsan különböznek-e. A SpeechTex rendszer szóhibaaránya szignifikánsan ($p < 0.01$) alacsonyabb volt, mint Google-é és Nuance-é, ami nem meglepő annak fényében, hogy a SpeechTex rendszer *in-domain* nyelvi adaton lett tanítva. Részletes eredmények a 3. táblázatban találhatóak, ebben fel vannak tüntetve mindhárom rendszer mindkét olvasási módon (normál, illetve az írásjelek diktálásával) elért szóhibaarányai. Jól látható, hogy a Google és Nuance rendszerek az írásjelek diktálása módban magasabb szóhibaarányal dolgoznak, ezek nyelvi modelljei noha tartalmazzák az írásjelek (pont, vessző, stb.) kiejtett alakjait mint homofónokat, tehát nem abban funkcióban és sorrendiségben mint ahogy azok az írásjeles diktálási módban történő diktáláskor szerepelnek. A SpeechTex rendszernél a normál olvasási mód eredményez magasabb szóhibaarányt, hiszen a nyelvi modell tartalmazza az írásjelek kiejtett alakját, és ezek elég gyakran fordultak elő a korpuszban ahhoz, hogy el nem hangzásuk rontsa a felismerési pontosságot.

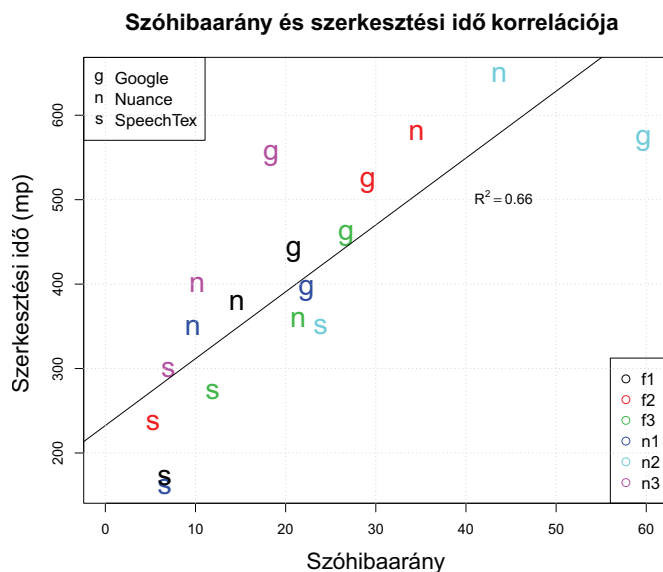
Szerkesztési időt tekintve azonban csak a SpeechTex rendszeré szignifikánsan ($p < 0.01$) alacsonyabb, mint Google-é és Nuance-é, ez utóbbi kettő közt nincs szignifikáns különbség.

Mint azt a 1. ábra mutatja, hogy a szóhibaarány és szerkesztési idő közötti korreláció szignifikáns és erős ($R^2 = 0.66$). Ezt árnyalja, hogyha az írásjelek diktálhatóságát mint faktort tekintjük; a 2. ábrán jól látható, hogy ugyanolyan szóhibaarány mellett az írásjelek diktálhatóságát lehetővé tevő rendszer alacsonyabb szerkesztési időt eredményez.

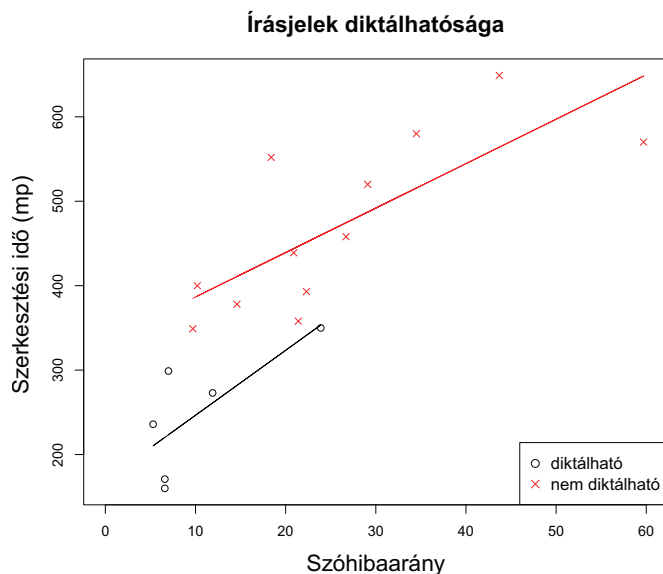
A 3.1. fejezetben leírt sikerességi ráta átlagosan 97,5% a Google, 98,9% a Nuance és 99,4% a SpeechTex rendszernél, és kizárólag a SpeechTex és Google közti különbség szignifikáns. Érdekes a sikerességi ráta összefüggése a szerkesztési idővel; azt találtuk, hogy fordított kapcsolat áll fenn: minél hosszabb a szerkesztési idő ($R^2 = -0,47079$), annál alacsonyabb a sikerességi ráta. Ez azt jelenti, hogy átlagosan több hiba marad egy több ideig szerkesztett kimeneti szövegben.

3. táblázat. Szóhibaarányok.

Résztevő	Normál			Írásjeles		
	Google	Nuance	SpeechTex	Google	Nuance	SpeechTex
n1	22.3	9.7	8.3	38.5	23.4	6.6
n2	59.7	43.7	40.8	51.4	33.3	23.9
n3	18.4	10.2	8.3	34.2	na	7.0
f1	20.9	14.6	11.2	36.2	19.3	6.6
f2	29.1	34.5	10.7	42.4	32.9	5.3
f3	26.7	21.4	14.1	40.7	21.0	11.9
Átlag	29.5	22.3	15.5	40.5	25.9	10.2
Szórás	15.2	13.9	12.5	6.0	6.6	7.0



1. ábra. Szóhibaarány és szerkesztési idő korrelációja.

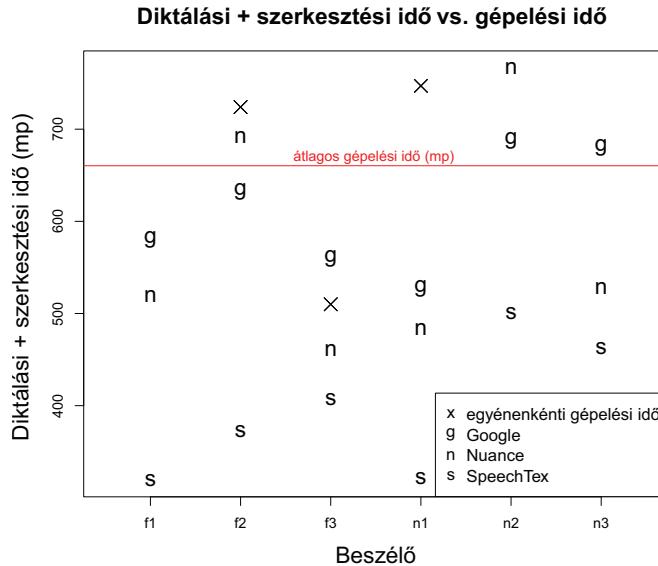


2. ábra. Szóhibaarány és szerkesztési idő az írásjelek diktálhatóságának függvényében.

4. táblázat. Diktálási tapasztalat és szerkesztési vs. gépelési idő becslése az automatikus beszédfelismerőn alapuló diktálás függvényében (nincs, ‘van’, ‘rendszeres’)

a diktálás és szerkesztés ... mint a gépelés	nincs	van	rendszeres
sokkal lassabb	1		
lassabb	1		
ugyanannyi		2	
gyorsabb			
sokkal gyorsabb			2

Végül a diktálással és szerkesztéssel eltöltött időt mértük össze ugyanazon szöveg begépelésének idejével - ilyen adat csak 3 résztvevőtől állt rendelkezésünkre. A 3. ábra mutatja, hogy néhány kiugróan magas együttes szerkesztési és diktálási időtől eltekintve, az együttes diktálási és szerkesztési idő rövidebb, mint a gépelési idő átlagosan. Az utólagos kérdőívből azonban az derült ki, hogy a diktálási tapasztalattal nem rendelkező résztvevők hosszabbnak érezték a diktálással és szerkesztéssel együttesen eltöltött időt, mint ugyanezen szövegbeviteli feladat gépeléssel való végrehajtását (l. 4. táblázat).



3. ábra. Diktálási és szerkesztési idő összevetése a gépelési időtartammal.

6. Összefoglalás

A beszédfelismerési kutatások középpontjában tipikusan a szófelismerési hiba csökkentése áll. Azonban az egyes speciális alkalmazásoknál, mint például a diktálás, a felhasználó számára közvetlenül nem a szóhibaarány, hanem elsősorban a diktálásra és javításra fordított idő csökkentése releváns. Tanulmányunkban az utóbbi célt tűztük ki. Egyrészt a számunkra közvetlenül hozzáférhető SpeechTex beszédfelismerési motor feladatára szabását végeztük el, másrészt a megoldásunkat összehasonlítottuk a lehetséges piaci alternatívákkal. A kimeneti szöveg jólformázottsága érdekében a fejlesztés során kiemelt hangsúlyt kapott az írásjelek diktálhatósága. Ezzel a funkcióval ismereteink szerint a vizsgálatok végzésekor nem bírtak a magyar nyelven ingyenesen hozzáférhető piaci termékek. Az összehasonlítást kontrollált körülmények között végeztük, kitüntetett figyelemmel a hibajavítási folyamatra. Az eredmények igazolták, hogy a lecsökkent szóhibaarány gyorsabb hibajavítással jár együtt. Ugyanakkor, tapasztalataink szerint az írásjelek diktálás során történő elhelyezése magát a hibajavítási folyamatot is gyorsította azáltal, hogy a felismerési kimenetben előforduló hibák detektálását megkönnyítette. Vagyis, az írásjelek diktálását lehetővé tevő megközelítés ugyanolyan szóhibaarány mellett alacsonyabb szerkesztési időt eredményezett. A kísérleteinkben résztvevő diktálási tapasztalattal nem rendelkező felhasználók azonban így is hosszabbnak érezték a diktálással és hibajavítással eltöltött időt, mint a szöveg begépelésének időtartama. Ennek egyik lehetséges oka, hogy az írásjelek diktálása szokatlan a felhasználó számára, hiszen a beszélt nyelvre ez nem jellemző. Megoldás lehetne az automatikus írásjelezés, de a legújabb kuta-

tásokban elért 50% körüli pontossága általános témakörben egyelőre nem valós alternatíva. Az automatikus írásjelezés pontossága kötött témakörnél alkalmazott gépi tanulási eljárással várhatóan jelentősen fokozható, ahogy a beszédfelismerési pontosság is magasabb kötött témájú korpuszon való tanítás esetén. Végül a hibajavítási felület ergonomikussá tétele és a felismerési hibák automatikus detektálása is lehetőségek a diktálás megkönnyítésében - ebben az irányban további kutatásokat tervezünk.

Köszönetnyilvánítás

Kutatásunkat a PIAC_13-1-2013-0234 (Patimedia) és KMR_12-1-2012-0207 (DIANA) projektek támogatták.

Hivatkozások

1. Tarján B., Nagy T., Mihajlik P., Fegyő T.: Magyar nyelvű, kísérleti e-mail diktálórendszer. In: IX. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2013), Szeged, Magyarország (2013) 21–28
2. Désilets, A., Stojanovic, M., Lapointe, J.-F., Rose, R., Reddy, A.: Evaluating Productivity Gains of Hybrid ASR-MT Systems for Translation Dictation. In: Proc. of IWSLT 2008, Hawaii, USA (2008) 158–166
3. Kolar, J., Lamel, L.: Development and Evaluation of Automatic Punctuation for French and English Speech-to-Text. In: Proc. of Interspeech 2012, Portland, Oregon, USA (2012) 1374–1377
4. Batista, F., Caseiro, D., Mamede, N. and Trancoso, I.: Recovering capitalization and punctuation marks for automatic speech recognition: Case study for Portuguese broadcast news. *Speech Communication*, Vol. 50, No. 10 (2008) 847–862
5. Vicsi K., Szaszák Gy., Németh Zs.: Prozódiái információ használata az automatikus felismerésben; mondatmondalítás felismerése. In: V. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2007), Szeged, Magyarország (2007) 69–80
6. D. Yu, M.-Y. Hwang, P. Mau, A. Acero, and L. Deng: Unsupervised learning from users' error correction in speech dictation. In: Proc. of Interspeech 2004, Jeju Island, Korea (2004) 1969–1972
7. Leijten, D.J.M., van Waes, L.: Error correction strategies of professional speech recognition users: Three profiles. *Computers in Human Behavior*, Vol. 26 (2010) 964–975
8. Jurafsky, D. and Martin, J.H.: *Speech and language processing. An introduction to NLP, computational linguistics, and speech recognition*, Englewood Cliffs, NJ: Prentice Hall (2000)
9. Steinberger, R., Eisele, A., Klocek, S., Pilos, S., Schlüter, P.: Dgttm: A freely available translation memory in 22 languages. In: Proc. of LREC 2012 (2012) 454–459
10. G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book*, version 3.4. Cambridge, UK: Cambridge University Engineering Department (2006)
11. A. Stolcke, Srilm – an extensible language modeling toolkit. In: Proceedings International Conference on Spoken Language Processing 2002, Denver, USA (2002) 901–904

12. M. Mohri, F. Pereira, and M. Riley: Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, Vol. 16, No. 1 (2002) 69–88