Nyílt forráskódú morfológiai elemző

Németh László*, Halácsy Péter* Kornai András**, Trón Viktor***

Kivonat A cikk a Szószablya projekt keretében kifejlesztett nyelvfüggetlen morfológiai elemző keretrendszert mutatja be. A MorphBase rendszer a Myspell helyesírás-ellenőrző programkönyvtár továbbfejlesztése. Új tulajdonságai, a többszörös affixumleválasztás, a kimeneti információ, az alternatív tövek és elemzések lehetősége, és az összetettszó-kezelés, alkalmassá teszik bonyolult agglutináló nyelvek helyesírás-ellenőrzésére, tövezésére, és morfológiai elemzésére. Az algoritmusok futásidőben feldolgozott nyelvspecifikus célra optimalizált erőforrásokat használnak (tő- és affixumtár), amelyek karbantartását és generálását nagyban megkönnyíti az erre a célra kifejlesztett HunLex előfeldolgozó.

1. Bevezetés

A 2003-ban indult Szószablya projekt [?,?] egyik célja egy olyan nyílt nyelvtechnológiai eszközkészlet elkészítése volt, amely bővíthető, más rendszerekbe könnyen integrálható és szabadon felhasználható. A projekt műszakilag olyan elemzőalgoritmusok kidolgozását tűzte ki célul, amelyek támogatni tudják tetszőlegesen komplex agglutináló alaktannal rendelkező nyelvek elemzését. A rendszer teljes architektúráját az 1. ábra szemlélteti. A HunTools eszközrendszer három futtatható komponense a Hunspell helyesírás-ellenőrző, a Hunstem tövező és a Hunmorph morfológiai elemző. Ezek mindegyike és az ezek alatt futó MorphBase alapkönyvtár és így az egész HunTools eszközkészlet természetesen nyelvfüggetlen, így a nevükben szereplő *Hun-* előtag a fejlesztés eredetére, nem pedig a magyar nyelvű erőforrásokra utal. A keretrendszer alapját képező elemzőalgoritmusok az Ispell helyesírás-ellenőrző család által használt módszer továbbfejlesztett változatai. A rendszer implementációja az Ispell technológián alapuló helyesírás-ellenőrző függvénykönyvtár, a Myspell kódjának továbbfejlesztésével történt, és több szószintű elemzőalgoritmust tartalmaz: helyesírás-ellenőrzés, tövezés és morfológiai elemzés automatikus vagy interaktív hibajavítással. A számos új képességgel felruházott alapkönyvtár az általánosabb MorphBase nevet kapta.

Az algoritmusok függvénykönyvtár formájában is elérhetők, amely a szoftvermodulokkal együtt a GNU LGPL licenc alatt szabadon felhasználhatóak.

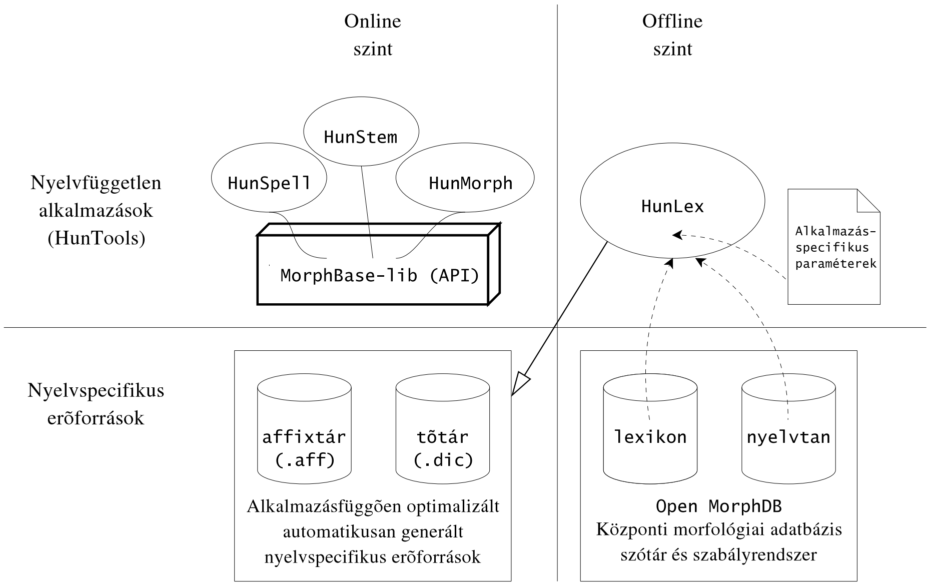
A Szószablya projekt kiemelt célja egy magyar nyelvű nyílt morfológiai elemző kifejlesztése volt. Az ehhez szükséges nyelvi erőforrások – magyar morfológiai

* Budapesti Műszaki Egyetem Média Oktató és Kutató Központ, {nemeth,halacsy}@mokk.bme.hu

** MetaCarta Inc., andras@kornai.com

*** International Graduate College, Saarland University and University of Edinburgh, v.tron@ed.ac.uk

szótár és szabályrendszer – előállítását és továbbfejlesztését nagyban képes segíteni a HunLex előfeldolgozó komponens [?]. Ennek a munkának az eredménye egy folyamatosan bővülő morfológiai adatbázis, amelyre épülő elemző rendkívüli lefedettséget és kielégítő pontosságú elemzéseket ad.



1. ábra. A Szószablya szóelemzési technológia felépítése

A nyelvtechnológia fejlődését mindig is komolyan befolyásolták a helyesírás-ellenőrzési alkalmazások. Ez a magyarban is így volt: a hétköznapi felhasználók számára fontos Helyes-e? jóval megelőzte a csak a szakértőknek érdekes HelyesLem és Humor alkalmazásokat [?]. A rendszer belső logikáját tekintve azonban azt látjuk, hogy ha megbízhatóbbá akarjuk tenni az elfogadás vagy elutasítás „egyszerű" dichotómiáján alapuló döntéseket, az alaktani tudás egyre alaposabb rendszerbe építésére van szükség. Jól mutatja ezt a Spell, Ispell, Myspell programcsalád fejlődése is.

A cikk további részében ismertetjük az Ispell technológia történeti hátterét és működésének alapjait (§2). Ezt követően tárgyaljuk a MorphBase algoritmusainak újításait (§3), valamint az egyes elemzési algoritmusok különbségeit (§4).

2. Affixumleválasztásos helyesírás-ellenőrzés

Első közelítésben a helyesírás-ellenőrzés leegyszerűsíthető egy gyakori és helyes szóalakokat tartalmazó halmazban való keresésre [?]. Az első helyesírás-ellenőrzésre használható, Les Earnest által a hatvanas években fejlesztett SPELL program szótára is csupán a leggyakoribb tízezer angol szóalakot tartalmazta[?].

Nyilvánvaló, hogy kizárólag a lista bővítésével teljes lefedettséget nem lehet elérni, de a szótáron kívüli (out-of-vocabulary, OOV) elemek forrása inkább a produktív alaktanban keresendő: a jól formált összetett szavak és toldalékolt alakok egy morfológiailag komplex nyelv esetében végtelen halmazt alkothatnak.

Earnest tanítványa, Gorin 1971-ben nemcsak a szótárat bővítette tovább, hanem a hatékonyabb Spell helyesírás-ellenőrző programmal bevezette a – még heurisztikus, azaz nyelvtanilag motiválatlan – szuffixumleválasztást.

Az általunk is használt, a tövek és a hozzájuk kapcsolható affixumcsoportok elkülönült tárolása miatt az erőforrások méretének jelentős csökkenését eredményező *affixumleválasztást* (affix stripping) mint egyszerű elemzési módszert[**?**] egészítette ki Ackerman a tövekhez csatolható affixumokat kódoló ún. *affixumkapcsolókkal* (affix flags). A jelentősen átírt program 1978-ban Ispell (ITS version of Spell) néven látott napvilágot, amelynek változatai ma is széles körben használatosak, és amely a MorphBase fejlesztés kiindulópontjának tekinthető. Az Ispell szellemű elemzési módszer élesen szétválasztja a nyelvfüggetlen elemző algoritmust a nyelvspecifikus erőforrásoktól, amelyek az adott nyelv töveit és toldalékolási szabályait szöveges állományok formájában adják meg.

A tőtár egykarakteres affixumkapcsolókat tartalmaz a tőtől perjellel elválasztva, amelyek az affixum-szabályok alkalmazhatóságát vezérlő privatív lexikai jegyeknek is felfoghatók:

```
kár/A
jár/AB
```

Az affixumállomány pedig definiálja a kapcsolókhoz tartozó affixumokat:

```
SUFFIX A om
SUFFIX B tam
```

A fenti definíciók a két szótári szón kívül a *károm*, *járom* és a *jártam* alakokat is generálják.

A Geoff Kuening által 1998-ban bevezetett *affixumtömörítésnek* (affix compression) köszönhetően egy affixumkapcsoló nem csak egy affixumot, hanem affixumok tetszőleges halmazát jelöli, így az affixumkapcsolók immár részparadigmákat engedélyező jegyként interpretálhatók. Tételezzük fel, hogy affixumdefiníciónk a következő:

```
SUFFIX A ba
SUFFIX A ban
SUFFIX A ból
```

Ekkor, a szótárban szereplő *vár/A* definíció a *várba*, *várban*, *várból* alakot egyaránt engedélyezi. Ez az affixumtáblán alapuló megoldás nagyszámú toldalékolási szabály redundanciamentes kódolását teszi lehetővé, miáltal az Ispell technológia hatékonyan alkalmazható az angolnál lényegesen komplexebb morfológiájú agglutináló nyelvekre is.

Az affixum-szabályok alkalmazhatóságát további feltételekkel korlátozhatjuk: az affixumállományban minden szabályhoz megadható még egy illesztési feltétel, továbbá egy (a tő széléről) levágandó karaktersorozat. Az alábbi példában a harmadik mező az affixum illesztése előtt a tő végéről levágandó karaktersorozatot tartalmazza (vagy nullát).[1] A negyedik mező a tő végén alkalmazott illeszkedési feltételt tartalmazza :

```
SUFFIX B 0 val   [óúv]
SUFFIX B 0 zal   [^s]z
SUFFIX B z szal  sz
```

Az első szabály szerint az *ó*, *ú* és *v* karakterre végződő tövek (amelyeknek van `B` kapcsolója) megkaphatják a *val* szuffixumot. A második engedélyezi, hogy azok a *z* karakterre végződő tövek, amelyek utolsó előtti betűje nem *s*, *zal* szuffixumot kaphatnak. A harmadik szabály szerint pedig az *sz*-re végződő szavak *szal* toldalékot kapnak, de a toldalék illesztése előtt levágásra kerül a tő végi *z* karakter. A levágásokkal az egyszerűsítő írásmódot használó összetételek (*sz+sz* → *ssz*), illetve a „hasonulások" (*lát* → *lássa*) alternatív tövek felvétele nélkül valósíthatók meg.

3. A MorphBase fejlesztés összetevői

A MorphBase fejlesztése egy olyan nyílt helyesírás-ellenőrző fejlesztéséből indult ki, amely képes volt a magyar nyelv komplex agglutináló morfológiáját, valamint a magyar helyesírás bonyolult rendszerét kezelni [?]. A MorphBase kódjának alapja a Myspell könyvtár, amely az Ispell technológia C++ nyelvű implementációja.[2] A Myspell előnyei, (i) az affixumleválasztást felgyorsító indexelés [?], (ii) a nyelvspecifikus erőforrások futásidőben történő beolvasása, (iii) a szálbiztosság, és nem utolsósorban, (iv) a teljesen szabad licenc, mind hozzájárultak ahhoz, hogy a fejlesztés alapjául válasszuk. Ezek a tulajdonságok a teljes HunTools programcsomagot is jellemzik, ezzel lehetővé téve a kód hatékony integrálását és újrafelhasználását.[3]

A MorphBase fejlesztés közül a legfontosabbak az (i) homonim tövek kezelése, (ii) a többszörös affixumleválasztás bevezetése, valamint (iii) a parametrizálható összetettszó-kezelés. Ezeket az újításokat az alábbiakban részletesen tárgyaljuk.

3.1. Homonimák

A *homonimák* kezelése a Myspellben nincs megoldva: egy karaktersorozathoz (tőhöz) pontosan egy affixumkapcsoló-halmaz rendelhető. Ez problémát jelent

[1] Prefixum esetén a tő elejére, szuffixumnál a végére vonatkozik a levágás.

[2] A Myspell könyvtár az OpenOffice.org nyílt forráskódú irodai programcsomaghoz készült és a korábbi zárt kódú ellenőrző-modult váltotta ki.

[3] A HunTools programcsomag C++ nyelven íródott, és a nyílt forrású fejlesztésekben használt standard segédeszközök (Make, jelenleg Automake, Autoconf) biztosítják a program platformfüggetlenségét.

a pontos helyesírás-ellenőrzés számára, hiszen így a kategoriálisan többértelmű tövekhez több homonim tő (lemma) jegyeinek unióját kellett rendelni. Ezzel azonban az adott tőformához ellentmondó affixumok is illeszkedhetnek egyazon szóalakban. Például a helytelen **elöle* alak is elfogadásra kerül az *öl* ige okán engedélyezett igekötő és az *öl* főnév okán engedélyezett névszói toldalék egyidejű megjelenése miatt. A MorphBase esetében megvan a lehetőség a homonimák elkülönítésére, így az igei prefixumok és a névszói szuffixumok egyidejű megjelenését pusztán a szótári besorolások le tudják tiltani. A tőszótár tehát támogatja a homonimák megadását:

```
öl/P  # ige
öl/S  # főnév
```

Így az affixumállományban szereplő igei és főnévi szabályok, pl.:

```
PREFIX P el
SUFFIX S e
```

egyazon tőre nem alkalmazódnak.

3.2. Rekurzív affixumleválasztás

Az Ispell az elemzés során csak egy prefixum- és egy szuffixumszabály alkalmazását engedi meg. Egyetlen szuffixum leválasztása szűk lehetőséget kínál az olyan agglutináló nyelvek kezelésére mint a magyar, hiszen az összes számításba jövő toldalékmorf-kombinációt mind egy-egy affixumszabálynak kell megfeleltetni. A teljes magyar inflexiós rendszer húszezer körüli kombinációt jelent, a produktív derivációs toldalékolás miatt viszont egy főnévi lemmának akár $10^3 - 10^6$ alakja is lehet, az igekötőket is figyelembe véve pedig újabb két nagyságrenddel nagyobb számot kapunk.Az affixumkombinációk tárolása tehát komplex morfológia esetén praktikus problémaként merül fel. Ennek a megoldására a MorphBase elemző algoritmusa többszörös affixumszabály-alkalmazást is megenged. Ez annyit jelent, hogy elemzéskor az affixumleválasztással feltételezett hipotetikus tövekről újabb affixumok választhatók le. A többszörös affixumleválasztás jelenlegi megvalósítása ugyan nem teljes rekurziót, csupán kétszeres szuffixumleválasztást takar, gyakorlatilag viszont már ezzel is négyzetesen csökkenthető a szuffixumok száma, így az erőforrások mérete, vagyis végső soron a program memóriaigénye. A magyar nyelvi erőforrás affixumállományában így sikerült a produktív képzőket korlátozások nélkül leírni, miközben a képzett alakok elhagyásával a tőszótár mérete is jelentősen csökkent.

Ennek a bővítésnek köszönhető az is, hogy a prefixumok már nem csak a tövekhez, hanem az affixumokhoz is köthetők. Mivel az alkalmazásuk feltételeként megkövetelhetnek egy affixumot, valójában circumfixumok implementálását is lehetővé teszik. Például a magyarban circumfixumként kezelendő a melléknévi felsőfokot kifejező *leg–bb* toldalékegyüttes: A *leg-* prefixum csak akkor kapcsolódhat a tőhöz, ha a *-bb* (megfelelő alakja) is kapcsolódik (**legpiros*, *legpirosabb*). Az ilyen circumfixumok kezelése eddig csak a szavak szótárban történő felsorolásával volt lehetséges.

A rekurzív affixumleválasztás implementálásával természetesen a nyelvspecifikus erőforrások formátumát is ki kellett bővíteni.[4] Az affixumdefinícióban két új mező jelenik meg. A kimeneti morfológiai információ (az elemző számára, 6. mező), illetve a „folytatási információ" (7. mező), amely az affixált alakra alkalmazható további affixumszabályok kapcsolóit tartalmazza.

```
PREFIX P   0 leg   .
SUFFIX R   0 ak    . [PLUR][NOM]
SUFFIX Q   0 bb    . [SUP]         PRN
SUFFIX Q   0 bb    . [COMP]        R
```

A példában a harmadik szabályban a `P` kapcsolók jelenléte engedélyezi a *leg-* prefixumot. Mivel a *leg-* prefixumra csak a *-bb* toldalékaffixumot tartalmazó minta hivatkozik, az első sor szabálya csak a harmadik sor alkalmazása „után" adhat helyes alakot és így az olyan alakok mint **legpiros* ki vannak zárva. A helyesírás-ellenőrzéshez már ennyi is elég volna, hiszen a *-bb* szuffixum viszont állhat a *leg-* nélkül (*pirosabb*), csakhogy a morfológiai elemzés számára ez nem állja meg a helyét, hiszen a *pirosabb* szóalak középfokú melléknévként elemzendő. A *-bb* toldalékmorf valójában homonim, és csak a felsőfok jelentésben „folytatható" a *leg-* prefixummal, sőt felsőfok jelentésben kötelező a *leg-* prefixum (e kölcsönös függés miatt nevezzük circumfixumnak). Azt, hogy egy szóalak csak tovább toldalékolva jelenhet meg, szintén egy kapcsoló adja meg (a harmadik sorban lévő `N`, ti. „nem tő"). Utóbbival tetszőleges olyan kötött tövek is megadhatók a szótárban, amelyek a továbbtoldalékolás szempontjából hasznosak, de nem jelenhetnek meg szabadon (pl. *lov*, *bokr*). A kötött tövek kezelését lehetővé tevő speciális kapcsoló szintén a fejlesztés során bevezetett újítás.

3.3. Összetett szavak kezelése

A számos további bővítés és új programparaméter közül az összetett szavak kezelését érdemes kiemelni. Az összetettszó-kezelés főbb tulajdonságai:

- megadható, hogy mely szavak szerepelhetnek szóösszetételben, akár csak az összetett szó első, vagy utolsó tagjaként, (a Hunspell esetében a legtöbb köznév ilyen, kivéve például a hónap- és napneveket),
- megadható a 6–3-as szabály (*kerékpárjavítással*, de *kerékpár-javítási*),
- megadható, hogy mely affixumok megléte esetén szerepelhetnek, illetve nem szerepelhetnek szóösszetételekben a képzővel ellátott szavak (*mérőléc*, de **mérendőléc*).

Tapasztalataink szerint a nyelvfüggetlen összetettszó-kezelés nem, vagy csak igen körülményesen valósítható meg: az összetett szavak helyesírása igen nehezen

[4] Ez a formátum kompatibilis a régi Myspell erőforrásokkal is.

parametrizálható a nyelvi erőforráson keresztül. Ezért a forráskódban elkülönítve jelenik meg egy az összetett szavak felismerését segítő osztály, lehetővé téve a modulnak a különböző nyelvekhez illetve célokhoz adaptált változatainak elkészítését.

4. Egy módszer, egy adatbázis, számos algoritmus

A helyesírás-ellenőrzéssel szemben a morfológiai elemzéshez vagy a tövezéshez nem elegendő a szóalakok elfogadásáról dönteni, hanem a releváns morfoszintaktikai kategóriák jelenlétét felismerve a bemeneti szóhoz kimenetként annotációt, tőindexet vagy lemmát kell rendelni. Emiatt az elfogadáshoz szükséges elemzéssel párhuzamosan a MorphBase elemző- és tövezőalgoritmusa a megfelelő kimenetkezelővel is kiegészül. A morfológiai elemzésnél minden szabályalkalmazás során az affixumhoz rendelt annotáció (kategóriacímke) regisztrálásra kerül, így sikeres elemzés esetén az algoritmus képes visszaadni a szó morfológiai elemzését.

A Myspell algoritmusa, akár a MorphBase helyesírás-ellenőrző algoritmusa az első elfogadott elemzés után nem keres továbbiakat, hiszen az elfogadáshoz ez nem szükséges. A tövezéshez és a morfológiai elemezéshez azonban a többértelműségek kezelése (nem a feloldásuk) alapvető követelmény, legyen az morf homonima (pl. *ár*, *fürdik*) vagy strukturális elemzési többértelműség (*értek*), így ezek az algoritmusok az összes alternatív elemzést meg tudják keresni. A teljes elemzés bevezetése egy helyen lehet korlátozva: a szavak összetett szóként való elemzését eszerint csak abban az esetben adja vissza a program, ha nem akad egyszerűbb elemzés. Például a *halász* elemzésénél nem kapjuk meg a szabályok szerint helyes *hal+ász* felbontást, mivel a szónak van más, nem összetett szavas elemzése is. Ez a módszer jól definiált nyelvfüggetlen szűrést eredményez, és tapasztalatunk szerint jelentősen csökkenti a felesleges többértelműségeket. Az olyan alakok, amelyeknek a homonim egyszerű elemzése mellett szükséges az összetett szóként való elemzése is (pl. *karóra*), a szótárba kerülnek, és így nem esnek áldozatul a korlátozásnak.

Az elemzés az affixumlevágásos módszerrel történik mindhárom szóelemző feladat esetében. A javaslattevés és hibajavítás képességétől elvonatkoztatva tehát az algoritmusok csak az alábbi két dimenzió mentén parametrizálhatók:

1. **elemzés teljessége**:
   (a) első elfogadott elemzésig (helyesírás-ellenőrzés, tövezés gyorsított indexeléshez)
   (b) korlátos többszörös elemzés
   (c) teljes elemzés (egyszerűbb elemzés esetén is adhat összetett szavas elemzést)
2. **kimenetfeldolgozás**:
   (a) nincs (helyesírás-ellenőrzés)
   (b) csak a tőtárban (tövezés)
   (c) mind a tövekhez, mind az affixum-szabály alkalmazáshoz (morfológiai elemzés)

A feladatokhoz szükséges minden egyéb konfiguráció a nyelvi erőforrások szintjén történik. Bár ez a feladat egyáltalán nem triviális, a HunLex előfeldolgozó segítségével egy közös adatbázisból automatikusan állíthatók elő a valósidejű alkalmazások számára különféleképpen optimalizált erőforrások.

5. Összefoglalás

Cikkünk általános tanulsága – ahogy ezt a MorphBase (a Hunspell, Hunstem, és Hunmorph programok alapja) is bizonyítja –, hogy a három legfontosabb szószintű elemzési feladat, a helyesírás-ellenőrzés, a tövezés, és a morfológiai elemzés egységes módszertannal kezelhető. Ez programozástechnikailag nem evidens, hiszen ugyanahhoz a bemenethez a helyesírás-ellenőrzés bináris (elfogad, elutasít) döntést, a tövező a tövet, a morfológiai elemzés pedig egy összetett, részint paradigmatikus (inflexió), részint szintagmatikus (deriváció) adat-struktúrát rendel. Az, hogy ezt a három problémát mégis egységes keretben érdemes kezelni, a tudományos közfelfogással is ütközik némileg, hiszen a helyesírás-ellenőrzést könnyű (lényegében listázással megoldható), míg a morfológiai elemzést nehéz, bonyolult algoritmusokat és nyelvészeti szakértelmet igénylő problémának szokás tekinteni.

A HunTools nyílt forráskódú programkönyvtárunk egyedülálló lehetőséget teremt magyar nyelvű szövegek elemzésére. A HunTools rendszer a GNU LGPL licenc alapján szabadon felhasználható, módosítható és nagyobb ipari rendszerekbe integrálható, akár önállóan futtatható szoftvermoduljait, akár a MorphBase függvénykönyvtár algoritmusait használva. A kompatibilis erőforrásformátumnak köszönhetően több, mint 40 nyelvhez használható helyesírásellenőrzőként, valamint szótövezőként (a morfológiai kimeneti annotáció hiányában morfológiai elemzőként nyilvánvalóan nem). Eszközünk alkalmazására különösen az Ispell technológia által mostohábban kezelt agglutináló nyelvek körében számítunk. Felhasználása mind kutatói, mind ipari körökben elkezdődött: a visszajelzések igazolják és egyben tovább növelik nyílt forrású fejlesztési modellünk sikerét.

Köszönetnyilvánítás

A Szószablya projekt az Informatikai és Hírközlési Minisztérium ITEM pályázatán nyert támogatással vált lehetővé. A program erőforrásait jelentős részben a MATÁV Rt. és az Axelero Internet biztosította. Fejlesztéseinkhez a Szószablya fejlesztőkön kívül a Magyar Ispell és a Szószablya levelezőlisták olvasóinak észrevételei is hozzájárultak. Segítségüket mindannyiuknak köszönjük.

Hivatkozások

1. B Dömölki. Algorithms for the recognition of properties of sequences of symbols. *USSR Computational & Mathematical Physics*, 5(1):101–130, 1967. Pergamon Press, Oxford.

2. Les Earnest. Machine recognition of cursive writing. *Information Processing*, 1963. Proc. IFIP Congress 1962, Munich.
3. Péter Halácsy, András Kornai, László Németh, András Rung, István Szakadát, and Viktor Trón. Szógyakoriság és helyesírás-ellenőrzés [word frequency and spell-checker accuracy]. In *Proceedings of the 1st Hungarian Computational Linguistics Conference*, pages 211–217. Szegedi Tudományegyetem, 2003.
4. Péter Halácsy, András Kornai, László Németh, András Rung, István Szakadát, and Viktor Trón. Creating open language resources for Hungarian. In *Proceedings of Language Resources and Evaluation Conference (LREC04)*. European Language Resources Association, 2004.
5. Németh László. Magyar Ispell – Válasz a Helyes-e?-re. In *IV. GNU/Linux szakmai konferencia*, pages 99–107. Linux-felhasználók Magyarországi Egyesülete, 2002.
6. Németh László. A Szószablya fejlesztés. In *V. GNU/Linux szakmai konferencia*. Linux-felhasználók Magyarországi Egyesülete, 2003.
7. James Lyle Peterson. *Computer programs for spelling correction: an experiment in program design*, volume 96 of *Lecture Notes in Computer Science*. Springer, 1980.
8. Gábor Prószéky and László Tihanyi. Humor – a morphological system for corpus analysis. In *Proceedings of the first TELRI seminar in Tihany*, pages 149–158, Budapest, 1996.
9. Trón Viktor. Hunlex - morfológiai szótárkezelő rendszer. In *II Magyar Számítógépes Nyelvészeti Konferencia*, 2004.