Konkordancia: írói szótár előkészítése számítógépen

A konkordancia: gépi termék. Jó előkészítéséül szolgálhat egy írói szótárnak — nem elkészítéséül, az továbbra is az emberre vár, de e l ő k é s z í t é s é ü l. Erről lesz szó az alábbiakban.

1. Kezdjük kitérővel, pontosabban a feladat megoldásának bizonyos értelemben vett ellenkezőjével: mit n e m szabad abban az esetben a számítógéppel csináltatni, ha költői szótárat, általánosabban szólva szó szintű szövegfeldolgozást akarunk vele előkészíttetni. Kérdésünket azért érdemes ilyen furcsán megközelíteni, mert egyrészt nagyon általános az alábbi hiba: tíz programozó közül nyolc vagy talán kilenc is így áll neki először a dolognak, ezt mind a nemzetközi, mind a hazai tapasztalat tanúsítja; másrészt igen tanulságos fényt vet a nyelvekre általában s a magyar nyelvre különösen.

Ha a szó szintű szövegfeldolgozás kérdése felmerül, akkor a programozó, attól függetlenül, hogy külön személy-e vagy a kutató nyelvészben élő másik szakember-e, körülbelül így kezd gondolkodni: Mi is egy szöveg — Arany János összes művei, a mai újság egy cikke, egy tankönyv stb. stb. —, ha az alkotó szavak (a szövegszók) szemszögéből nézem? Hosszabb-rövidebb betűsorok egymásutánja. De hiszen betűk vagy számok, az mindegy, azt a szöveget, hogy „Ég a napmelegtől a kopár szik sarja", mondjuk, így is elképzelhetem magam elé: 1012 01 200126190917091231251701 stb., ahol is az „é" betűnek a 10-es szám, a „g" jelnek a 12-es szám, az „a"-nak a 01-es stb., stb. felelt meg rendre. Hát akkor egy szöveg szótárát összeállítani nem más, mint ezeknek a számoknak a tárát — felsorolását — elkészíteni, valamilyen sorrend szerint rendezve benne e „számokat". — Szakítsuk itt meg a programozó gondolatmenetét, csupán jelezve, hogy ezen a ponton, talán annak hatására is, hogy a programozó tudatában felkódlik úgyszintén a „gyakorisági szótár" neve vagy fogalma, de talán azért is, mert olyan közel van hozzá a s z á m o l ó g é p — egy újabb végzetes ötlet születik, ilyesféle: „. . . és számláljuk is meg az azonos »számokat«, rögzítsük, hányszor fordult elő a 1012, a 01 stb., stb." Látszólag minden eddigi gondolat, a legutóbbit is beleértve, teljesen helyes. A kutató azonban a vizsgált valósággal sakkozik: ilyen megnyitás után menthetetlenül mattot kap a valóságtól — a nyert eredmények nem fognak a valóság megismeréséhez vezetni. Ezt mutatjuk ki a következő bekezdésekben.

E gondolatmenet eredményeképpen a következő listákat fogja kiadni a gép, attól függően, hogy a programozó ötletei merre hajladoztak: (i) *monda* 6, *az* 51, *úr* 8, *Jónásnak* 3, *kelj* 3, *fel* 1, *és* 62, *ment* 1 stb. Terminológikusan szólva (vö. Nyr. 88:76—82): vette az egyes szövegszókat abban a sorrendben, ahogy előfordultak a szövegben, szóalakok szerint csoportosította és megszámolta őket. Még ugyanezen a változaton belül maradva, hamar rájön a programozó arra, hogy a 6 *monda* mellett volt 1 *monda:* szóalak is (ti. ugyanaz, csak kettősponttal a végén), a *Jónásnak* 3 szóalakján kívül van még 2 *Jónásnak:* stb.; csinál egy külön programrészt, amely a nyers szóalakokat meg-

tisztítja ezektől az elemektől és eredményül már valamivel reálisabb adatokat kap. (A fenti számok egyébként teljesen pontosak a Jónás könyve alapján — saját konkordancia-eredményeinket „rontottam vissza" az itt ismertetendő programnak megfelelően: ha ezen, (i) program szerint dolgoznánk, akkor ezeket az eredményeket kapnánk.) Jegyezzünk meg még annyit, hogy miként a *monda* és a *monda:* két külön szóalak a gépnek, éppen úgy például az *úr* és az *ur* is az: ha tehát az adatrögzítés során véletlenül egyszer-kétszer rövid *u*-val lyukasztották ezt a szóalakot vagy az alapul vett kiadás olykor — indokolatlanul vagy indokoltan: például mert a kéziratban is így volt, mondjuk metrikai okokból, tájnyelvi hatásként stb., stb. — rövid *u*-val hozza, akkor ez szigorúan más, az *úr*-tól különböző szóalak lesz, másutt kerül felsorolásra, külön számoltatik stb. (ii) *a* 141, *ad* 1, *addig* 1, *adtak* 1, *ahol* 2, *ahova* 1, *akar* 1, *akará* 1, *akarta* 1, *akartad* 1, *akárhogy* 1, *akárki* 1, *aki* 7, *akik* 1 stb. Tehát: ugyanaz, mint az (i) alatti, csakhogy a szóalakok itt már betűrendben állnak. (iii) 1. *a* 141, 2. *s* 91, 3. *és* 62, 4. *az* 53, 5. *Jónás* 29 (ezenkívül van még 2 *Jónásból*, 5 *Jónásnak*, 1 *Jónások*, 3 *Jónást* — összesen 40 szövegszóban testesül meg a „Jónás" lexéma), 6. *nem* 25, 7. *hogy* 20 stb. Tehát: ugyanaz, mint a (ii) alatti, csakhogy itt már az egyes szóalakok nem betűrendben állnak, hanem előfordulásuk számának csökkenő sorrendjében. Örül a programozó: „kész gyakorisági szótár!"

Az öröm korai, a nyelvi anyag megtréfálja a kutatót. Az ilyen listák erősen sántítanak többek között a következő okokból:

a) H o m o n i m i a. Az ÉrtSz. közel 60 000 címszavából mindössze néhány százalék a homonima-indexszel ellátott elem, aránylag több található tőszókincsünkben. Ám a homonimák ravaszak legalább két szempontból. Egyrészt: a leggyakoribb szavak között gyakrabban fordulnak elő mint a szótárban; tehát szövegfeldolgozás esetén nagyobb problémát jelentenek, mint a szótár alapján gondolhatnánk. Íme, mindjárt *az az* mai nyelvünkben persze leggyakrabban határozott névelő, de sokszor mutató névmás; az *egy* névelő vagy számnév stb. S ha erre még azt is mondhatná valaki, hogy egyes tekintélyes szótáraink is együtt kezelik mint többszófajúságúakat ezeket a lexénákat, szétválasztásuk tehát mesterséges — a szintén igen gyakori *ki*-re például, már egyáltalán nem mondható ugyanez: vagy egy — éppen különírt, erről l. alább — igekötő, vagy névmás (már ne is szóljunk arról, hogy talán nem érdektelen: milyen névmás az adott esetben). Másrészt: a homonimák a szövegben olykor a szótártól függetlenül is feltűnnek, amennyiben valamely ragos-jeles alak esik egybe egy kiinduló alakúval, egy hasonló ragos-jeles alakkal stb.

b) R a g o z á s. A toldalékolás miatt egy-egy lexéma más-más szóalakjai természetesen külön vannak felsorolva mind a három verzióban, legföljebb csak más-más sorrendben. Így nagyon természetes, hogy az *akar, akará, akarta, akartad* fenti példánkban külön-külön leszámoltatott, ezekre a számokra szükségünk is van — de itt meg „nem végzett teljes munkát" a gép: végül mégiscsak nekünk kell összeadással megállapítanunk, hányszor is fordul elő az *akar* lexéma. Ez persze egy olyan piciny szöveg esetében, mint a Jónás könyve, nem számít — de nagyobb szövegek esetében? A ragozáson belül az alakrendi kiegészülés külön (magyarban szerencsésen nem túlságosan nagy) problémát jelent: a *van* lexéma előfordulásait például egyáltalán nem olyan egyszerű még a betűrendes listáról sem összeállítani, mint az *akar*-ét. Vö.: *vagyok, van, legyen, lehet, lesz, volt* stb. — ezek mind többé vagy kevésbé más szókezdetek, tehát még a (ii) listán is több helyről kell őket összeszednünk, a (iii) meg e szempontból teljesen összezavarná a helyzetet. Hiszen ahhoz, hogy a (iii) alapján megállapítsuk e valóban gyakori lexéma szövegbeli előfordulási számát, tudnunk kellene, hogy az egyes alakok hányszor fordultak elő, s majd azon előfordulási szám alatt keresni őket: a Jónás könyvében pl. 5 *vala*, 4 *volna, volt*, 3 *vagyok*, 2 *lesz, lett, lőn, vagy* (ez meg szépített adat: a

valóságban a (iii) szerinti listán a 8-szor előfordulók között állna ez a szóalak: ebből 6-szor a *vagy* kötőszót fedné) és így tovább.

c) I g e k ö t ő. Magyarban (németben stb.) az elváló igekötő összezavarja az (i)—(iii) listát, a homonimiától függetlenül is. Van valahol, mondjuk 26 *ki*, és abból valahogy megállapítottuk, hogy — mondjuk — 14 valóban nem névmás. Ugyanakkor előfordul ebben a szövegben több száz ige, közöttük *ki* igekötősek is: mármost melyikhez adjuk hozzá ezek közül azt a 14 *ki*-t úgy, hogy a gép által számlált 18 *megy*-ből csak 16 legyen, viszont a 3 *kimegy*-ből 5, mert a *ki* kétszer éppen ehhez a tőhöz járult és így tovább. A (iii) lista alapján a magyarban (németben stb.) még azt sem tudjuk megmondani, hogy hány lexéma is volt az adott szövegben — a különírt „valami" esetleg igekötő, esetleg valóban önálló határozószó (a németben: prepozíció); az egyes igék előfordulási száma is megállapíthatatlanul irreális: egyes igekötős igékből kevesebbet, más, igekötőtlen, igékből többet mutat fel listánk, mint amit a szöveg valóban tartalmazott. És ezzel körülbelül itt is a matt, amit a vizsgált anyag adott a kutatónak: felsejlenek előtte egyes mennyiségi arányok (hiszen minden gyakoriságuk ellenére is — nem minden homonimia, nem mindenütt vannak igekötők stb.) — de azután megint eltűnnek, annyira pontatlanok, hogy alig-alig lehet velük valamit kezdeni. Hányszor éri a kutató ilyen csalódás!

Az *a)—c)* pontokban itt éppen csak példaképpen mutattunk rá azon okok némelyikére, amelyek miatt az (i)—(iii) listák többé-kevésbé használhatatlanok; az *a)—b)* okok ezen belül elég általánosak (bár természetesen távolról sem univerzálisak), a *c)* specifikusabb.

2. Az eddig vázolt megközelítés(ek) hibája látszólag az, hogy a programozó nem vette kellően figyelembe a nyelvi anyag sajátosságát. A „szám-modell" (ti. az írott szövegnek számokkal teleírt laphoz való hasonlítása) nem elég erős, nem adja vissza kellően a természetes nyelvi szöveg sajátosságait, abból szinte csak egyetlen, külsődleges momentumot ragadva ki (hogy ti. a szöveg egymástól helyközökkel elválasztott jelsorokból áll). Valóban, ez a közvetlen hibaforrás.

Ám emögött valami más húzódik meg; vagy ugyanezt kissé más oldalról is nézhetjük. Nevezetesen: a programozó vitette magát a könnyen programozhatóság szellőjével. Az **1.** alatti változatokat igen könnyű programozni; ilyen feladatokat programozóoktatás során akár öncélúan is fel lehet adni, mint szép gyakorlatot, hamar sikerélményt hozót. „Beolvasni egy szöveget annak elejétől valamely »vége« jelig" — ez egy utasításblokk. „Ebből kiválasztani az első számot (később: a következő számot, space-től space-ig)" — egy másik. „Összehasonlítani a beolvasottat sorban minden számmal" — ez is remek: mondjuk, a beolvasott számból rendre kivonja a többi számokat, s ahol kivonás eredménye nulla volt, ugyanazt a számot (esetünkben: szóalakot) találta. És így tovább: az első vizsgált szóalak mellé annyi strigulát húz be magának a gép, ahány ugyanolyat talált; az egyszer már valamely szóalak kapcsán megtalált, strigulázott szövegszókat kipipálja magának, hogy a következő átfutáskor meg se álljon mellettük. (A „kipipálás" különféle módokon történhet. Például úgy, hogy az adott szám — szó — előjelét negatívra változtatja és a program eleve úgy épül, hogy először minden szám pozitív, majd pedig csak a pozitív számokat kell összehasonlítgatni-kivonogatni.) Amikor azután előálltak olyan rekeszegyüttesek, melyek mindegyike egy-egy szóalakot és egy-egy számot (ahányszor ez a szóalak előfordult) tartalmaz, akkor ezzel különféle dolgokat lehet csinálni: a fenti (i) esetében sorba kiíratni úgy, ahogy előfordultak, a (ii) esetében előbb a szavak nagysága szerint rendezni (= ábécébe rakni: az *a* a legkisebb betű, az *á* a rákövetkező és így tovább) és csak aztán kiíratni; a (iii) esetében az előfordulás gyakorisága szerinti számok alapján rendezni. Még a jó feladatokban szükséges buktató

is megvan itt, amelyen azután könnyen lehet segíteni: az ábécé szerinti „nagyságrend" nem ugyanolyan, mint a számokkal leírt nagyság szerinti rend: az *ablak* „kisebb" (előbb következik), mint az *az* névelő, jóllehet — ha csak úgy egész egyszerűen számoknak nézzük a betűket, akkor — az *az* kevesebb számjegyből állván eleve kisebb szám, mint az *ablak*.

Mindezt elmondva még engem is elkap a pedagógus lelkesedés (ezúttal: a programozást oktató lelkesedése), hogy milyen jó gyakorlat ez, milyen szépen tanítható, milyen érdekesen megoldható. És itt van a kutya elásva: erre nem szabad gondolni a programkészítés legelején, amikor egyáltalán a kutatási stratégiát dolgozzuk ki. Annál fontosabb ez az érzék-érzés később, amikor a helyes stratégia alapján a taktikát kimunkáljuk: elegancia, gépi idő, ez utóbbi kapcsán egyáltalán a megoldhatóság sokszor ezen áll vagy bukik. Még egyszer visszatérve korábbi hasonlatunkra: az **1.** szerint programozók emlékeztetnek arra a kezdő gyermeksakkjátékosra, aki a lépések ismeretének örömében igyekszik minél hosszabbakat lépni, vezérével és futárjaival rögtön megnyitás után az ellenség soraiba hatolni minden komolyabb, távolabbi meggondolás nélkül.

A programozásnak output-orientáltan kell indulnia: pontosan tudnunk kell, mit is akarunk a géptől, mennyit akarunk tőle (kezdő kutató rendszerint sokkal többet kér a kelleténél — a gép szívesen ki is adja neki mindazt, amit megkíván; ám a kiadott adattengerben elmerül a megrendelő: már megint matt). Világosan meg kell lennie a kutatási stratégiának, ennek kell alárendelni a program taktikáját. Ha csak lehet (ha volt ilyen előzmény), gondosan tanulmányozni kell: az e m b e r hogyan oldott meg korábban ilyen feladatot. A gépi megoldás menete nem lesz okvetlenül ugyanilyen — de tudni mégis kell róla: az ember is gép, ha távolról sem olyan jó a mechanikus feladatok megoldására, mint az általa létrehozott „igazi" gépek. Itt tehát valóban szükség van a nyelvi tények ismeretére, az eddigi kutatások tapasztalataira — például a már elkészült költői szótárakra, az esetükben alkalmazott lexikográfiai módszerek emlékezetbe idézésére.

3. Dimidium facti, qui bene coepit, habet. Ez ugyan a fentiekre is érvényes, de itt jelen sorainkra akarjuk vonatkoztatni: miután jól kezdtünk azzal, hogy a rosszat elmondtuk, ezzel több mint a felén túljutottunk a dolognak: ezek után igazán röviden elmondható, hogyan is kell esetünkben eljárni. Mi közvetlenül és rögtön nem szótárt akarunk kapni; amikor az ember-nyelvészek előttünk szótárakat készítettek, a szövegből nem közvetlenül és rögtön a lexémákat halászták ki, sőt mégcsak nem is az egyes szóalakokat. Amikor a nyelvész céduláz, kiírja a kérdéses szó környezetét, például azt az egész mondatot vagy tagmondatot, amelyben a szó előfordult, aláhúzva benne az ott számára érdekes elemet. Amikor a lexikográfusok nagyban dolgoznak, a nyomdától durván szólva annyi példány krúdát kérnek, ahány szó egy lapon van, az azonos lapokon sorba aláhúzzák az egyes szövegszókat, ezzel magkapva azok környezetét is. A cédulákat (az egyes krúda-lapokat) azután betűrendbe rakják az aláhúzott elemek szerint. Így szépen felgyűlnek az adatok az egyes lexémákra, már ahogy azok a különféle szóalakokban, a különféle kontextusokban realizálódnak. Ezenkívül természetesen mind a kézzel írt cédulán, mind a nyomdai lapon (ez utóbbin részben implicit formában) ott van a pontos forrásmegjelölés is: ez az adat itt és itt fordult elő.

Ezek után tekintsük meg az 1. ábrát — egy lapot Ady Szent Margit legendája c. költeményének konkordanciájából. Nem más ez, mint a lexikográfus cédulatára betűrendbe szedve: egy-egy sor egy-egy cédula és a kérdéses szó nem aláhúzással van kiemelve, hanem azzal, hogy két helyköz áll előtte. Íme, a kérdéses szavak (szövegszók): *jött, Ki, Királyi, királyi, klastromba, kósza, könnyes* . . . és az e lapon legutolsók (legalsók): *törékeny, trubadúr, udvaron, úr*. Vagyis ez a lap éppen *j*-től *ú*-ig kezdődő címszóalakokat tartalmazott. Látjuk az egyes címszóalakok környezetét is: „Ő nem *jött:*

egy csöndes álom-lovag." (A + jelet még külön is beteszi a gép, ezzel jelezvén mindenütt a mondat végét.) Vagy, a következő sorban (kihagyom a távolabbi, ott olvasható környezetet): „. . . Margitot, *Ki* ott halt meg . . .". És így tovább. Minden egyes sor végén áll még egy-egy szám: az jelzi az adat pontos előfordulási helyét. Az itt mindig ismétlődő „12" a vizsgált vers sorszáma (az autentikus kiadásban ui. ez a Vér és arany kötet A magyar messiások c. ciklusának épp a tizenkettedik költeménye); az ezt kötőjellel követő három szám annak a mondatnak a sorszáma, ahol e költeményen belül az adat áll. Tehát, például: a *jött* e vers 10. mondatában, a *Ki* a 11.-ben fordul elő. Cédulázó célunknak megfelelően egyébként a pontosvesszőt is mondatvégnek vettük: általában elég szokott lenni, ha egy pontosvesszővel elválasztott tagmondattól (-ig) írjuk ki a kontextust.

És kész, ezzel tulajdonképpen mindent elmondtunk a konkordanciáról. Nem más az, mint a szöveg minden egyes szavának ábécébe szedett listája, kontextussal és locus-megjelöléssel együtt. „Konkordanciának" — eléggé esetlenül — azért hívják, mert az egymással „megegyező", azonos szóalakok kerülnek benne egymás alá, egy csoportba. Magyarul tulajdonképpen szövegszó-tárnak kellene nevezni; ennél tartalmilag kevésbé megfelelő, de érthetőbb elnevezés lehetne a szóalaktár, amennyiben az azonos szóalakok képeznek benne egy-egy összefüggő csoportot, az „azonos"-ba ezúttal beleértve a homonim alakokat is. Az egyes „címszók" környezete programunkban tetszőleges hosszúságú lehet. Sőt, a gép a munka megkezdése előtt kiírja nekünk a hozzá csatlakoztatott írógépen: „Hány betűt balról, hányat jobbról?"; miután kapott két számot (például: 20, 30 = 'húszat balról, harmincat jobbról'), még megkérdi: „Egy lapon hány sor legyen?" — és csak akkor indul be, ha ezt a számot is megkapta. Akkor azután így, mindig azonos tükörrel, paginálva adja ki a bevitt szöveg szótárát (a pagináció az ábrákról lemaradt).

Ez utóbbi elegancia, a kiírt eredmény esztétikus és nyomdakész volta persze már nem elengedhetetlen tartozéka egy szóalaktárnak — konkrétan a mi programunk, a Görgey Eszter és Jékel Pál készítette, dolgozik ilyen intelligensen és szépen. Hadd mutassak rá a GE—JP program (a programokat készítőjük monogramja szerint szoktuk hívni) még egy egyedi sajátosságára. Tessék megnézni felülről a 3. sort: vajon miért áll annak a legelején egy magányos „,t. +"? A program tulajdonképpen megtette volna a magáét, ha a *Királyi* címszó környezeteként csak a megfelelő betűmennyiségnyit írja ki, így: „. . . fehér Margito" — és kész, többet a tükör nem enged. Ám ilyenkor, ha a sor elején még maradt üres hely — ezt persze külön ellenőriznie kellett a gépnek —, oda még beír annyit a mondat legvégéből, amennyi belefér: itt a hiányzó „t" betűt és a mondatzáró pontot. Ugorjunk néhány sorral lejjebb, a *Legendák* címszóhoz: ott meg a sor végén maradt még hely, arra hát kiírta a gép az adott mondat elejét. (Íme, a ti/tok: . . . *Legendák* szűzét, fehér Margitot. +) (Az esztétikus külsőn kívül az ilyen filológiai apróságok jelentik a költői elem egy piciny megnyilvánulását a programban; ezeket látva szoktunk itt büszkén „JP-iskoláról" beszélni.) E finomságok láttán talán nem is kéri számon tőlem az olvasó, hogy például miért sorszámmal jelöltük a verset és miért nem, mondjuk, a címének a rövidítésével — az eredeti, GE-program e helyt valóban három elemes volt, abból az első betűket tartalmazott, nekem azonban így jobban megfelelt a dolog, itt tehát direkt durvítottam a programon. Ha megvan a jó stratégia és költők a programozók — akkor mindent meg lehet csinálni.

Arról talán más alkalommal, hogy e szóalaktárak felől hogyan lehet továbbmenni a költői szótárak és a gyakorisági szótárak felé; meg arról is, hogy milyen lenne egy *a tergo* szóalaktár, melyből szintén készült már nálunk néhány.

Papp Ferenc