HTTPS://DOI.ORG/10.53644/MKI.OSI.2023.163

MINTAEVOLÚCIÓS ALAPÚ JELENTÉSAZONOSÍTÓ ELJÁRÁS

TÓTH LORÁND LEHEL – HOSSZÚ GÁBOR

KIVONAT: Az írásokkal (általánosabban mintarendszerekkel) készült, megfejtetlen feliratok (általánosabban jelszekvenciák) jelentésének meghatározása a jelszekvencia által megtestesített szimbólumszekvencia megtalálását jelenti. A SID jelentésazonosító eljárás a jelszekvencia jelentését legjobban megadó szimbólumszekvenciát távolságmetrikák és vektorműveletek használatával keresi meg. Ennek érdekében egy adott mintarendszerbe tartozó szimbólumok glifjei és a vizsgált jelszekvenciabeli jelek topológiai jellemzői alapján keresi meg egy szótár-adatbázisból a megfelelő szimbólumszekvenciát. A szótár-adatbázis megfelel a jelszekvencia létrejötte korának és a jelszekvencia által megtestesített szimbólumszekvencia feltételezhető nyelvének. Az eljárás figyelembe veszi, hogy egy időbeli evolúciót mutató mintarendszer szimbólumaihoz tartozó glifek egyes alakváltozatai a mintarendszer evolúciója során feledésbe merülhettek. A SID eljárás teljesítménye egy valós megfejtési eseten került bemutatásra.
KULCSSZAVAK: írásinformatika, jelentésazonosítás, mintaevolúció, mintarendszer, számítógépes paleográfia, székely-magyar rovás, távolságmetrika, topológiai jellemző

Bevezetés

A régészek által talált régi feliratok jelentős részének nincs olvasata, még akkor sem, amikor a keletkezésük korát egyéb módszerekkel sikerül meghatározni. Ennek oka az írásanyag (fa, kő, tégla, papír stb.) romlásán kívül az, hogy egy adott írásban használt grafémák glifjei (rajzolatai) az idők során módosulnak. A cikk egy ismeretlen jelentésű régi feliratok (általánosan jelszekvenciák) megfejtésére kidolgozott új eljárást mutat be. Az itt bemutatott kutatás a nehezen olvasható jelszekvenciák statisztikai módszerekkel történő megfejtésére összpontosít, az adott mintarendszer (esetünkben írás) szimbólumaihoz tartozó glifváltozatok és a megfejtetlen felirat jeleinek geometriai-topológiai jellemzői alapján.

Az általunk végzett vizsgálatokban a *mintarendszer (pattern system)* alatt a szimbolikus kommunikáció egyik formáját értjük, pl. az írás egy mintarendszer. A mintarendszer alkotórészei a szimbólumok, a szintaktikai szabályok és az elrendezési szabályok. A *mintaevolúciós vizsgálatok (pattern evolution research)* az időbeli evolúciót mutató mintarendszerek kutatása. Az írásinformatika a mintaevolúciós vizsgálatok azon része, amely egy speciális mintarendszerrel, az írásokkal foglalkozik. Az írásinformatikai kutatások hosszútávú célja egyrészt a világban feltárt nagyszámú megfejtetlen írásemlék értelmezése, illetve megfejtése, másrészt az, hogy a kutatókat segítse az írások evolúciós kapcsolatainak feltárásával (Hosszú 2014a, 2014b, 2019). Az írásinformatikának a régi feliratokkal foglalkozó részét *számítógépes paleográfiá*nak nevezzük.

Ha a mintarendszer egy írás, akkor a mintarendszert alkotó *szimbólum* mintarendszerbeli típusai a hangértékkel rendelkező *graféma,* a jelentéssel rendelkező *tamga* és a konkrét jelentés nélküli, díszítő funkciójú *díszjel.* A graféma lehet egy hangot jelölő betű, egy képjel (pl. egyiptomi hieroglif), írásjel (pl. felkiáltójel), ligatúra (betűk összevonva), számjegy stb. A szimbólumnak többféle tulajdonsága van, ezek egyike a *glif*je, ami vizuális vagy egyéb módon érzékelhető megjelenése. Itt a vizuálisan érzékelhető gliffel rendelkező szimbólumokkal foglalkozunk, amelyek glifje topológiai jellemzőkkel *(topological attributes)* leírható (Pardede et al. 2012). Ha a szimbólum egy graféma, akkor van átbetűzési értéke és hangértéke. Az átbetűzési érték a graféma helyettesítése latin

vagy görög betűvel, amelyet < és > jelek között szokás megadni. A *hangérték* a graféma által megjelenített, rendszerint fonéma értékű hang jelölése, amelyet fonemikus átírásnál / és / jelek között adunk meg. A fonemikus átírásnál nem a hajdani kiejtés hangtani értelemben pontos megjelenítése, hanem az egyes grafémák által jelölt fonémák azonosítása a cél. A *fonéma* egy elvont nyelvi egyed, egy adott nyelvben jelentésmegkülönböztető erővel bíró egység, a nyelv legkisebb tagolási egysége. A fonéma nem hang, hanem nyelvészeti elvonatkoztatás. A különbözően ejtett beszédhangokat a beszédészlelés fonémaszintjén azonosítjuk megfelelő fonémaként (Gósy 2004, 245.).

A *szimbólumszekvencia (symbol sequence)* a szimbólumok sorozata (pl. egy szó, egy mondat vagy egy hosszabb szöveg), amely megvalósulva (pl. leírva) *jelszekvenciát (graph sequence)* alkot. Ha a vizsgált mintarendszer egy írás, akkor a jelszekvencia pl. egy felirat vagy egy dokumentum. A szimbólumszekvencia egyik esete a *grafémaszekvencia,* feltéve, hogy a benne szereplő szimbólumok grafémák. A feliratokban szereplő, önálló vizuális szerepű topológiai alakzat általánosítása a *jel (graph),* amely a jelszekvencia alkotórésze. A jelekből álló jelszekvencia az ennek megfelelő szimbólumszekvencia megvalósulása, így egy jel egy megfelelő szimbólum jelszekvenciabeli megvalósulása. Egy jelszekvencia (pl. egy felirat) megfejtése egy megfejtetlen jelszekvenciához egy adott mintarendszerbe tartozó szimbólumszekvencia megtalálását jelenti. Egy szimbólumszekvencia vagy speciálisan egy grafémaszekvencia vizuálisan azonosítható megfelelője a glifszekvencia, amit a szekvenciába tartozó szimbólumok (speciálisan grafémák) glifjeinek felhasználásával kapunk. Egy glifszekvencia adott technológiával történő megvalósítása adja a jelszekvenciát, amit akkor, ha a szóban forgó mintarendszer egy írás, *felirat*nak nevezünk.

Az ismert grafémák glifjét és a feliratbeli jeleket egyaránt topológiai tulajdonságokkal lehet leírni, ezekből topológiai jellemzővektorok *(topological attribute vectors)* képezhetők. Topológiai tulajdonság lehet például egy körszerű hurok, ferde szakasz, függőleges szakasz, kereszteződés. E topológiai jellemzővektorokat távolságmetrikák és vektorműveletek segítségével dolgozzák fel annak érdekében, hogy megtalálják egy ismeretlen jelentésű jelhez az alakra leghasonlóbb ismert grafémák glifjeit úgy, hogy a vizsgált felirat értelmes szöveget adjon ki. Ennek alapja egyrészt az, hogy az ismert grafémákhoz nem

csak glifek, hanem hangértékek is tartoznak; másrészt a megfejtéshez a vizsgált jelszekvencia keletkezési korának és nyelvének megfelelő szótáradatbázis kerül felhasználásra, amelyben a szavak hangértékeikkel vannak eltárolva (Tóth–Hosszú 2019).

Az eddigi vizsgálatok azt mutatják, hogy a topológiai jellemzők helyes meghatározásával és a megfelelő távolságmetrikák alkalmazásával olyan jelentésazonosító eljárás *készíthető, amely figyelembe veszi, hogy egy írás* szimbólumaihoz (a vizsgált esetekben grafémákhoz) tartozó glifváltozatok az írás evolúciója során feledésbe merülhettek, így olyan jelszekvenciák (speciálisan feliratok) is megfejthetők, amelyekben a szereplő jelek mára feledésbe ment glifeket jelenítenek meg. A cikk először a módszer elméleti hátterével, majd a kidolgozott eljárás leírásával, a vizsgálati eredmények ismertetésével, végül a következtetések levonásával foglalkozik.

Háttér

Számos kutatás jelent meg a mintafelismerés, az optikai karakterfelismerés (*Optical Character Recognition,* OCR), írásfelismerés, régi feliratok megfejtése témakörében; a nemzetközi szakirodalomban található könyvek, cikkek, folyóiratok és konferenciák sokasága igazolja a téma fontosságát. A gépi tanulási eljárásokat széles körben használják a karakterfelismerés területén, ha nagy mennyiségű adat áll rendelkezésre a képzési halmazok összeállításához, ez a feltétel azonban nem mindig teljesül. Jó eredményeket értek el a kézzel írt bengáli számjegyek felismerésében a konvolúciós neurális hálózatok segítségével (a bengáli az indiai szubkontinens egyik fő beszélt nyelve, sőt Banglades első és hivatalos nyelve, Rahman et al. 2019). Az ókori feliratok vizuális felismerésére szolgáló különböző megközelítések összehasonlítása során 14 560 felirathoz kapcsolódó 17 155 fényképen végzett vizsgálatok azt mutatták, hogy a jellegfelismeréshez használt Fisher-vektor (Sanchez et al. 2013) és konvolúciós neurális hálózat jellemzőinek egyetlen képi reprezentációban való kombinálása az esetek több mint 90%-ában a lekérdezett feliratok helyes felismerésére vezetett (Amato et al. 2016).

Érdekes terület a régi feliratok digitalizálása és megfejtése 3D modellezési algoritmusokra támaszkodva. Barmpoutis et al. (2010) újszerű keretrendszert javasoltak a feliratok hatékony 3D rekonstrukciójára és a rekonstruált felületek statisztikai elemzésére. Egy *shape-from-shading* eljárást alkalmaztak a feliratos felületek alakjának 3D-s rekonstrukciójára, ehhez azokat egy-egy karaktert tartalmazó kisebb doboz alakú régiókra szegmentálták. Ezeket a karaktereket azonos karakterek vagy szimbólumok csoportjaiba sorolták, majd minden karakterhez egy-egy betűformákból álló atlaszt hoztak létre. Az atlaszok segítségével végezték a feliratok jeleinek automatizált elemzését. Ez a keret hatékonyan használható a glifek egy feliraton vagy feliratsorozaton belüli változatainak tanulmányozására. Egy másik kutatás keretében a sérült ókori feliratok automatizált rekonstrukciójára és vizualizációjára kidolgozott olyan algoritmust mutattak be, amely hibrid megközelítéssel a 2D és a 3D elemzési eljárások előnyeit egyesíti (Sapirstein 2019).

A textúramodellezéshez és textúrafelismeréshez bevezettek egy sokoldalú és hatékony keretrendszert, amely a lokális bináris mintákon számított forgásinvariáns attribútumok családján alapul. A topológiai tulajdonságminta *(topological attribute patterns)* alapú, hatékony textúramodellezési keretrendszer a topológiával kapcsolatos tulajdonságokat veszi figyelembe (Nguyen et al. 2016).

Az elmúlt évtizedben a számítógépes történeti nyelvészet új hulláma jelent meg, amelynek módszerei közé tartoznak a genetikai rokonság automatikus felmérése, az automatikus rokonnyelv-felismerés, a filogenetikai következtetés és az ősállapot-rekonstrukció. Ezekkel egy nyelvcsalád filogenezisének átfogó képe tárható fel (Jäger 2019, Rama et ali. 2018). A kihalt nyelvek automatikus megfejtésére kidolgozott modell alapja az, hogy számos nyelvi intuíciót statisztikai keretbe foglalnak (Snyder et al. 2010). A mára kihalt sémi nyelvre, az ugaritire alkalmazva a modell az ugariti szavak 60%-át összekapcsolja a héber rokon értelmű szavaival, és 30 betűből 29-et helyesen rendel hozzá a héber megfelelőikhez. Céljuk eléréséhez karakterszintű összehasonlításokat és statisztikai módszereket alkalmaztak.

A régi feliratokban alkalmazott mintafelismerési és adatbányászati technikák egyik területe a régi írások közötti kapcsolatok feltárása, beleértve az egyetlen gyökérírásból való lehetséges közös eredetüket is. Daggumati–Revesz

(2019) adatbányászati technikákat mutatnak be, amelyekben konvolúciós neurális hálózatokat és támogató vektorgépeket használnak nyolc különböző régi írásjelben található szimbólumpárok közötti vizuális hasonlóság mértékének megállapítására. Hosszú–Kovács (2016) eredményeket értek el a régen használt írások kapcsolatainak feltárásában. Ők egy gépi tanulási megközelítést mutattak be a régebbi korokban használt írások szimbólumainak egyes glifjei közötti fenetikai kapcsolatok feltárására, figyelembe véve az egyes glifváltozatok topológiai tulajdonságait és az írások evolúciója során az írások szimbólumaihoz tartozó glifek fejlődésében bekövetkezett átalakulásokat. Különböző klaszterelemzési módszereket alkalmaztak a régi írások szimbólumaihoz tartozó glifek hasonlósági csoportjainak meghatározására az írások közötti fenetikai kapcsolatok feltárása érdekében.

A régi feliratok készítőinek azonosítása gyakran szükséges a régészeti és történelmi kutatásokhoz, mivel segít felismerni a felirat tartalmának eredetét. Egy erre a célra szolgáló általános módszertan a mintafelismerés, a képfeldolgozás és a matematika módszereit tartalmazza (Rousopoulos et al. 2011): egy feliratban megjelenő írás szimbólumainak megvalósításait használja, és bizonyos kritériumok alapján összehasonlítja azokat. Újszerű statisztikai kritériumokat dolgoztak ki annak eldöntésére, hogy két vizsgált felirat készítője vajon ugyanaz a személy-e vagy különböző személyek készítették.

A régi feliratok megfejtése nehéz, főleg, hogyha a benne szereplő jelek által megjelenített szimbólumok glifváltozatai vagy akár a teljes szimbólumok egy része mára elfelejtődött. Ezért kutatásunk célja egy olyan jelentésazonosító eljárás kifejlesztése volt, amely képes régi, nehezen olvasható, ismeretlen jelszekvenciák azonosítására, megfejtésére. Az evolúciós alapú jelentésazonosítás jelentős részben eltér az optikai karakterfelismerés (OCR) szokásos feladatától (Chaudhuri et al. 2017). Míg az OCR esetében egy mintarendszer szimbólumainak tipikus (idealizált) gliffel (rajzolattal) rendelkező alakját rendszerint ismertnek tételezhetjük fel, s egy feliratban található képi információt kell megfeleltetni valamelyik ismert grafémának (Chen et al. 2004), addig az evolúciós alapú jelentésazonosítás során a feliratban található képi információt úgy kell valamilyen grafémához rendelni, hogy a graféma tipikus glifje maga nem ismert, vagyis a jelentésazonosítás a kérdéses szimbólum fejlődése során egy

valaha használt glifváltozat felismerésére és azonosítására koncentrál. Ugyanakkor az evolúciós alapú jelentésazonosítás és az OCR számos közös módszert alkalmaz, ilyenek például a glifszegmentálás és a jellemző-kiválasztás. Az OCR-módszereket hatékonyan használják a szöveg lokalizálására és szövegfelismerésre a videó- és képfeldolgozási alkalmazásokban (Chen et al. 2004).

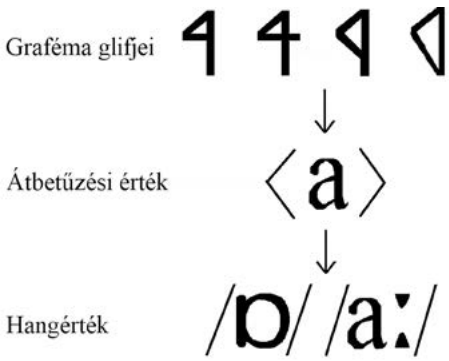
A kifejlesztett módszert a XV–XVII. századi székely-magyar rovás írással írt, egyes esetekben grafémahiányos, grafémahibás, vagy ismeretlen jeleket tartalmazó írások megfejtésénél teszteltük (Hosszú 2013a), de elvileg alkalmazható más írásokra, sőt használata kiterjeszthető egyéb mintarendszerekre is. Az elvégzett kutatás nehézségét az adta, hogy a vizsgált írás esetén az írásemlékekként tekinthető feliratok korlátozott számban állnak rendelkezésre (Tóth et al. 2015).

Módszer

A SID algoritmus

A kifejlesztett módszer neve SID (*inScription Identification,* 'felirat azonosítás'); jelenleg még csak egyszavas feliratokra alkalmazható. Az eljárás az ismert szimbólumok (a vizsgált problémák esetén grafémák) glifjei, valamint a jelszekvenciát (feliratot) alkotó jelek összehasonlítására épül (Hosszú–Kovács 2016; Hosszú 2017, 2021). Az ismert grafémák glifjét és a jelszekvenciabeli jeleket egyaránt *topológiai tulajdonság*okkal írjuk le. Egy tulajdonság lehet pl. zárt hurok, függőleges vonal, vízszintes vonal, végpont stb. E topológiai tulajdonságokból *jellemzővektor*okat képzünk, így minden ismert glifhez, valamint ismeretlen jelhez definiálunk egy-egy jellemzővektort, amelyet elmentünk egy adatbázisba. A grafémák ismert glifjei, hangértékei és átbetűzési értékei is szerepelnek az adatbázisban. A SID ezenkívül tartalmaz egy szótáradatbázist a megfejtendő felirat korának és feltételezett nyelvének megfelelő szavakból összeállítva. Ebben a szótáradatbázisban az egyes szavak hangértékeikkel szerepelnek (Tóth–Hosszú 2019). A SID jelenlegi szótáradatbázisa az egyes szavaknak a ragozott alakjait is külön-külön adatbázisbeli bejegyzésként tartalmazza.

A SID eljárása két részből épül fel, az első a *SID-Előfeldolgozó* és a *SID-Fő* algoritmus (Tóth et al. 2016b). A SID-Előfeldolgozó a jelszekvenciát alkotó jelek jellemzővektorainak az adott mintarendszerhez (íráshoz) tartozó grafémák ismert glifjei jellemzővektoraitól valamilyen mérték szerint számítható távolságát minimalizálva a feliratbeli jelekhez összegyűjti az adott jelhez hasonló glifeket (vagyis az összes lehetséges rokon glifet) a grafémák ismert glifjeinek adatbázisából, így minden jelszekvenciabeli jelhez több ismert glifet is találhat mint megfejtési jelöltet. Az így kiválasztott adatbázisbeli glifek minden esetben valamilyen grafémához tartoznak, ezért ezekhez *átbetűzési érték* és *hangérték* is tartozik (Hosszú 2013b). Különböző grafémáknak is lehet azonos átbetűzési értéke és azonos hangértéke, ahogy azt az 1. ábra is szemlélteti. Az 1. ábrán látható példa a székely-magyar rovás <a> graféma glifváltozatait, hangértékét és átbetűzési értékét mutatja be.



1. ábra: A glif – átbetűzési érték – hangérték megfeleltetés

A SID-Előfeldolgozó első lépése az ismeretlen jelentésű jelszekvencia egyes jeleihez tartozó topológiai jellemzők adatbázisba történő bejegyzése. Egy adott mintarendszerhez tartozó szimbólumok ismert glifjeit leíró topológiai jellemzővektorokhoz hasonló jellemzők különböző korábbi kutatásokban már szerepeltek (Das et al. 2012, Bag et al. 2011, Tirandaz et al. 2017, Zaghloul et al. 2011). Egy glif topológiai jellemzői olyan geometriai tulajdonságok (zárt hurok, függőleges vonal, vízszintes vonal, végpont stb.), amelyek vizuálisan felismerhetők és azonosíthatók.

A SID-Előfeldolgozó a vizsgált jelszekvenciához meghatározandó szimbólumszekvencia alaki megjelenéséül javasolt glifszekvenciákat hoz létre, amelyeket a jelszekvenciabeli jelekhez leghasonlóbb adatbázisbeli grafémák ismert glifjeinek kombinációjából képez. Ez az eljárás a talált leghasonlóbb glif és a megfelelő vizsgált feliratbeli jel egyezőségét nem garantálja.

A SID-Előfeldolgozó következő lépésként a glifszekvenciákból átbetűzésiérték-szekvenciákat hoz létre, és kiküszöböli a duplikációkat, majd a glifszekvenciákat lecseréli átbetűzésiérték-szekvenciákra. Erre azért van szükség, mivel több glif tartozhat azonos átbetűzési értékhez, így a glifszekvenciáknál kevesebb átbetűzésiérték-szekvencia jöhet létre. Az így létrehozott átbetűzésiérték-szekvenciákból kombinációkat képez, és végül az átbetűzésiérték-szekvenciákat lecseréli hangértékeikre. Ezáltal hangérték-szekvenciák jönnek létre, amelyekben szintén kiküszöböli a lehetséges duplikációkat. Több hangérték tartozhat azonos átbetűzési értékhez, így az átbetűzésiérték-szekvenciák számánál, valamint a glifszekvenciák számánál is több hangérték-szekvencia jöhet létre eredményként. Az algoritmus e lépéseket azért hajtja végre, hogy a kombinációk számát a futásidő-optimalizálás végett a legalacsonyabban tudjuk tartani.

Ezáltal létrejött az összes lehetséges hangérték-szekvencia, amelyek közül a ténylegesen létezettek kiválaszthatók a vizsgált jelszekvencia készítési korának és az ehhez használt nyelvnek megfelelő szótáradatbázisból. A szótáradatbázis találatai a SID-Előfeldolgozó kimenetén jelennek meg, illetve a SID-Fő algoritmus bemenetére kerülnek. A SID-Előfeldolgozó lényegében meghatározza a bemeneti jelszekvencia (felirat) megfejtésére a legrelevánsabb jelöltek halmazát, de még nem tudható, hogy ezek közül melyik lesz a legjobb találat.

A SID-Fő algoritmus kvantitatív módszereket alkalmazva meghatározza a vizsgált jelszekvenciához (felirathoz) leginkább hasonló glifszekvenciákat és a hozzájuk tartozó hasonlósági szintjüket. Bemenetén megkapja a lehetséges szimbólumszekvenciákat (grafémaszekvenciákat, esetünkben szavakat), amelyeket egyenként dolgoz fel a következő műveletek során. E lépésekben a SID-Előfeldolgozó lépései köszönnek vissza fordított sorrendben. A SID-Fő algoritmus lecseréli a szótáradatbázisból származó szavak hangértékét átbetűzési értékükre, majd az átbetűzési értékekből átbetűzésiérték-szekvenciákat képez kombinációkkal. Az így létrejött átbetűzésiérték-szekvenciákat a hozzájuk tar-

tozó glifekkel helyettesíti, így glifszekvenciák formálódnak. Ennek során az egyes átbetűzési értékekhez tartozó összes lehetséges gliffel létrehoz glifszekvenciákat. Ezen glifszekvenciák mindegyike olyan grafémaszekvenciát jelent, amelyekhez tartozó hangérték-szekvenciák találhatók a szótárban, vagyis ezek mindegyike lehetséges megoldás.

Utolsó lépésként a generált glifszekvenciák egyes elemeihez tartozó jellemzővektorokat összehasonlítja az eredeti felirat (jelszekvencia) jeleihez tartozó topológiai jellemzővektorokkal. A hasonlóságmérést a szövegek osztályozására használják, és a klaszterelemzés is ismert módszer a közös jellemzők megtalálására és két dokumentum közötti hasonlóság kiszámítására (Lin et al. 2014). A SID algoritmusa a hasonlóságmérést egy ismert glif és egy ismeretlen szimbólum között végzi el a geometriai-topológiai jellemzővektoraik felhasználásával. Ennek során a SID két távolságmetrikát, a *Hamming-távolság*ot és az *euklideszi távolság*ot alkalmazza, mindkettőnek széleskörű felhasználási tapasztalata létezik (Shehu et al. 2015, Cunderlik–Burn 2016).

A Hamming-távolság két egyenlő hosszúságú sztring (szimbólumszekvencia) közötti távolságot úgy határozza meg, hogy megadja azon pozíciók számát, ahol a két sztringbeli szimbólumok eltérnek egymástól. Így a minimális számú olyan helyettesítések számával egyenlő, amelyek az egyik sztringnek a másikba való átalakításához szükségesek (Hamming 1950).

A d_{xy} két sztring euklideszi távolságát írja le, ahol x jelöli a megfejtendő felirat jelszekvenciájának egyik jelét, az y pedig az algoritmus által generált glifszekvenciák ugyanazon helyiértékén levő glifjét, lásd (1). Az x_j a jelhez tartozó topológiai jellemzővektor j-ik elemét jelöli, míg az y_j a vizsgált glifhez tartozó topológiai jellemzővektor j-ik elemét jelöli. Az n az általunk definiált topológiai jellemzők száma.

$$d_{xy} = \sqrt{\sum_{j=1}^{n} |x_j - y_j|^2}. \tag{1}$$

A SID-Fő eredményül a legrelevánsabb glifszekvenciákat adja vissza a különbözőségi értékeik szerint növekvő sorrendben. Az algoritmus kidolgozása közben számos algoritmusgyorsító, és pontosságjavító technikát,

módszert is kidolgoztunk és alkalmaztunk, melyek segítségével az algoritmus futási idejét csökkentettük, ugyanakkor növeltük a pontosabb találat valószínűségét. A továbbiakban e módszereket ismertetjük a továbbiakban (vö. Tóth et al. 2021).

Egy átbetűzési érték különböző glifekhez és különböző hangértékekhez tartozhat (Hosszú 2013b). Az átbetűzési értékeket esetünkben dimenziócsökkentési célokra használtuk. A beszédfelismerő rendszerek számára a fonéma-grafféma megfeleltetés egyik megközelítését (Basson–Davel 2013) mutatja be.

Következő lépésként a SID-előfeldolgozó algoritmus a glifszekvenciák halmazát a rokon glifek szimbólumonkénti kombinációjaként hozza létre. Ez a lépés azért fontos, mert a leginkább rokon glifek és a vizsgált szimbólum teljes egyezősége nem garantált. Meg tudjuk határozni a feldolgozandó leginkább rokonértelmű glifek mennyiségét. Az ismert glifek halmaza a tesztfuttatások során időnként változhat a különböző kezdeti bemeneti paraméterbeállítások megválasztása alapján.

Mielőtt a korábban kiválasztott betűkészletből kombinációkat készít, az algoritmus egy készleten belül kicseréli a betűket az átbetűzési értékeikre, majd egyesíti őket, kiküszöbölve a duplikációkat. Ennek eredményeképpen ugyanaz a glifkészlet különböző glifalakokat tartalmazhat azonos hangértékekkel. A kombinációk végrehajtása ezekkel a kisebb halmazokkal átbetűzésiérték-szekvenciákat hoz létre. Végül az algoritmus az átbetűzési értékeket hangértékekre cseréli, mivel egynél több hangérték tartozhat ugyanahhoz az átbetűzési értékhez. Ezáltal a hangértékek összes lehetséges kombinációjából álló hangérték-szekvenciák jönnek létre, amelyek most már kereshetők a felirat feltételezett nyelvének és a keletkezési korának megfelelő szótáradatbázisban.

A szótáradatbázisban található találatok lesznek a SID-Előfeldolgozó kimenete és a SID-Fő bemenete. Ebben a fázisban határozzuk meg, hogy a találatoknak a bemeneti szimbólumsorozat legrelevánsabb megfejtését kell tartalmazniuk, de a hasonlóság mértékét még nem ismerjük. A SID-Fő feladata ennek a hasonlósági szintnek a meghatározása kvantitatív leírás segítségével, és a megfejtetlen szimbólumsorozat számára a leghasonlóbb glifváltozathoz tartozó jellemzővektor kiválasztása. Így megvan a potenciális szavak halmaza, amelyet a következő műveletekkel egyenként feldolgozunk.

Az algoritmus a hangértékeket az átbetűzési értékeikre cseréli, ezért az átbetűzési értékeket a glifjeikkel helyettesíti. Ezzel a módszerrel az algoritmus a glifváltozatokból generálja az összes lehetséges glifszekvenciát. Végül a generált glifszekvenciák topológiai jellemzővektorait két alaptípusú, de nagyon hatékony távolságmetrika, a Hamming- és az euklideszi távolságmetrika (Shehu et al. 2015, Cunderlik–Burn 2016) segítségével összehasonlítja a szimbólumsorozat topológiai jellemzővektoraival, és a SID-Fő algoritmus kimeneteként kiválasztja a hasonlósági értékek növekvő sorrendjében a legrelevánsabb glifszekvenciák halmazát.

Osztályozási módszerek

Az *osztályozási módszerek* bevezetésének célja a SID-Előfeldolgozó gyorsítása volt. Az ismert glifek és a megfejtendő felirat jeleit klaszteranalízissel csoportosítottuk, felhasználva egyes topológiai jellemzőiket (Tóth et al. 2016a). Ennek a csoportosítási módszernek az elnevezése *T-osztályozó* (T a „topológiai" szóból). Minden alkalommal, amikor egy új grafémát vagy egy ismert grafémához tartozó új glifet veszünk fel az adatbázisba, a klaszterelemzési módszert végre kell hajtani, hogy megtaláljuk a megfelelő csoportot, ahová a glif kategorizálható. A T-osztályozó mellett egy másik kategorizálás, a *V-osztályozó* került a SID eljárásba, amely a T-osztályozótól eltérően heurisztikus módszeren alapul. Ennek lényege, hogy a T-osztályozónál alkalmazott szigorú topológiai leírást mellőzve az adott feliratbeli jel vagy ismert glif összképe alapján kerül osztályokba és ezáltal a kézírások okozta esetlegességeket sikerül részben kiszűrni (Pardede et al. 2016).

A V-osztályozó kimenete esetenként eltérhet a T-osztályozó kimenetétől. Ha új graféma vagy egy ismert grafémához tartozó új glif kerül az adatbázisba, akkor annak a T-osztályozó és V-osztályozó besorolási mezői lesznek meghatározva. Ha a felhasználó beállítja ezen paraméterek egyikét a felhasználói felületen, akkor egy vizsgált feliratban lévő még ismeretlen jel első körben csak az azonos (T- vagy V-) osztályba tartozó ismert glifek halmazával kerül összehasonlításra. Ez egy gyorsítási módszer, amely relevánsan csökkenti a SID futási idejét, és csak a SID-Előfeldolgozóban került implementálásra.

Az osztályozási módszerek hátránya, hogy ha egy grafémát vagy egy glifjét rosszul osztályoznak, a SID-Előfeldolgozó nem találja meg a lehetséges egyezéseket a szótáradatbázisban, ezáltal a SID-Fő sem kapja meg a bemeneti adatokat, végeredményül pedig az algoritmus egyetlen találatot sem ad. A negatív eredményt, amely e módszer alkalmazása során esetlegesen előfordul, a *Levenshtein-távolságmetrika* alkalmazásával küszöböltük ki.

Levenshtein-távolságmetrika

A SID-Előfeldolgozó a generált hangérték-szekvenciák (S^{σ}) és a szótár-adatbázis szavainak (W^{σ}) összehasonlítása során utolsó lépésként a *Levenshtein*-távolságmetrikát alkalmazza (Putra–Supriana 2015). A $d_L(S^{\sigma},W^{\sigma})$ távolságmetrika két karakterlánc összehasonlítását végzi, és a távolság definíciója a karaktereken elvégzett csere, törlés, vagy beszúrás műveletek azon minimális száma, amely ahhoz szükséges, hogy az S^{σ} karakterláncot átalakítsuk W^{σ} karakterlánccá (Levenshtein 1966, Zhao–Sahni 2019). A függvényt a (2) képlettel írtuk le.

$$R^{\sigma} = (d_L(S^{\sigma}, W^{\sigma}) \leq s_{th}),\ \acute{e}s\ R^{\sigma} \subset W^{\sigma} \tag{2}$$

A szótáradatbázisból azokat a szavakat (grafémaszekvenciákat, R^{σ}) választjuk ki, amelyeknél ez az érték elér egy előre meghatározott hasonlósági küszöbértéket (s_{th}). Ez az s_{th} küszöb határozza meg a megengedett eltérések számát a generált hangérték-szekvencia S^{σ} és a szótáradatbázisban szereplő szavak (grafémaszekvenciák, W^{σ}) között. Ez a SID szoftver felhasználói felületén átállítható. Minél nagyobb a beállított küszöbérték, annál több lehetséges megoldást jelentő szóval tér vissza a SID-Előfeldolgozó, még akkor is, ha azok nem elég relevánsak; ellenkező esetben kevesebb pontos találat kerül a SID-Előfeldolgozó kimenetére.

A Levenshtein-távolságmetrika alkalmazásának célja a SID-Előfeldolgozó algoritmus találati valószínűségének növelése. A vizsgálati eredmények bizonyították, hogy a SID-Előfeldolgozó használható találatokat adott még azokban az esetekben is, amikor a generált hangérték-szekvenciák (S^{σ}) nem egyeztek

meg pontosan a szótár-adatbázisban található szavak W^{σ} egyikének a hangértékeivel sem.

Körmódszer

A *körmódszer (circle method)* egy geometriai-topológiai alapú eljárás, amelyet a topológiai tulajdonságok mellett a glifek, valamint a jelek pontosabb leírására fejlesztettünk ki (Tóth–Hosszú 2019). A SID pontosságát tovább javítottuk az általunk kidolgozott körmódszerrel, amely három koncentrikus körnek a poláris koordináta-rendszer segítségével a glifekre, ill. jelekre való szerkesztéséből adódik, ezáltal keresztmetszeti pontokat generál a vizsgált glifek és feliratbeli jelek vázaival (Stauffer et al. 2019, Kanimozhi 2012).

A körmódszer alapja az, hogy egy saját fejlesztésű Matlab szoftver használatával leképezzük a glifek, valamint a jelek vázát, megkeressük ezek geometriai középpontját, majd három különböző sugarú koncentrikus kört szerkesztünk rájuk. Ezekután rendre összeszámoljuk a koncentrikus körök és a glif/jel vázak metszéspontjait, és ezeket az értékeket háromelemű vektorban tároljuk el. A szoftverfutásnál opcionálisan ezzel a háromelemű vektorral egészíthetjük ki a topológiai jellemzővektorokat.

A tesztelések eredményei azt mutatták, hogy a körmódszer által biztosított jellemzővektor használata pontosabb megfejtési eredményeket adott a nehezen olvasható jelekből álló feliratok megfejtése során, azon az áron, hogy a SID feldolgozási ideje növekedett (Tóth–Hosszú 2019).

Találatszám

A *találatszám (number of matches)* paraméter bevezetésének célja az algoritmus futási idejének és a találati valószínűségnek az optimalizálása volt. Ez a paraméter határozza meg azt, hogy a feldolgozandó ismeretlen jelhez mennyi leghasonlóbb glifet vegyen figyelembe a SID. Ha ez a paraméter nagy értékre van állítva (*>5*), akkor nő a valószínűsége, hogy az adatbázisban rendelkezésre álló ismert glifek halmazából kiválasztható legalább egy az ismeretlen jelhez hasonló glif. Ez a találati valószínűséget növelő módszer fordítottan arányos

a feldolgozási idővel, mivel a vizsgált glifszekvenciák száma exponenciálisan nő a szóba jöhető ismert glifek halmazának növekedésével. Ezért a találatszám paramétert érdemes kisebb értéken tartani (*<5*), különösen akkor, ha hosszabb szavakat szeretnénk megfejteni (Tóth et al. 2016b).

Erősségszint

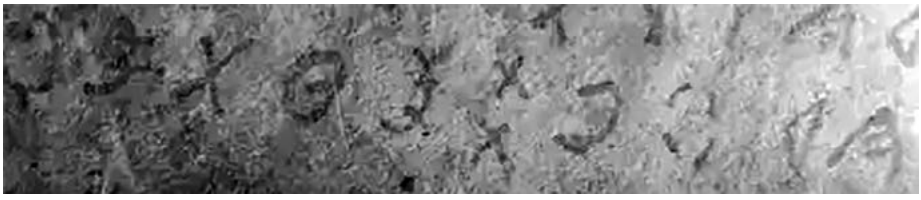
Az *erősségszint (robustness level)* paramétert a SID algoritmus futási idejének csökkentésére és minél több lehetséges megfejtés megjelenítésére vezettük be. Az *erősségszint* paraméter jelentősen csökkenti a SID-Fő feldolgozási idejét azáltal, hogy a megoldásra jelölt glifszekvenciákat (amelyek a szótáradatbázisban megtalálható szavakat írnak le) egy előre meghatározott minta szerint kiválasztja a már generált glifszekvenciák teljes kombinációs halmazából. Az előre meghatározott minta (szám) írja le, hogy a generált teljes glifszekvencia-halmaz hányadik eleme kerüljön feldolgozásra.

Csak az előre szűrt glifszekvenciák kerülnek vizsgálatra, vagyis a hasonlósági metrikák csak az így kiválasztott glifszekvenciákon hajtódnak végre. Az *erősségszint* paraméternek egy természetes számot lehet beállítani *1* és *99* között. Ha ennek értéke *1*-re van állítva, akkor minden egyes generált glifszekvencia topológiai jellemzővektorai rendre össze lesznek hasonlítva a bemeneti jelszekvencia jeleinek topológiai jellemzővektoraival. Ha ez a paraméter egy *x* értéket kap (*1<x<99*), akkor az új minta alapján csak minden *x*-edik generált glifszekvencia lesz analizálva. Ebből már látszik, hogy előfordulhat, hogy a végeredményben a kiválasztott legközelebbi glifszekvencia nem lesz a legpontosabb leképezése a vizsgált, megfejtendő jelszekvenciának, ugyanakkor ez a körülmény a jelszekvencia által megtestesített szimbólumszekvencia, vagyis esetünkben a keresett szó megtalálásában nem okoz gondot.

Az *erősségszint* paraméter értéke kevésbé befolyásolja a legjobb megfejtések megtalálását, ugyanakkor lehetőséget ad különböző glifszekvenciákkal és hasonlósági távolsággal megadott találati szavak megjelenítésére a kimeneten.

Eredmények

Az összetett SID-algoritmust megvalósító szoftverbe különböző gyorsítási és találati pontosságot növelő eljárásokat építettünk be. A kidolgozott módszerek hatékonyságát egy valós példán keresztül mutatjuk be. A Patakfalvi-bibliában található, 1776–1785 közti időszakból való kétoldalas székely-magyar rovás írással készült felirat (Hosszú 2013b, 250–251.) egy egyedi jeleket tartalmazó egyszavas részletét (2. ábra) olvasat nélkülinek tekintve a jelentésazonosító szoftvernek sikerült olvasatot adnia a vizsgált részletre. A vizsgált írásemlék a székely-magyar rovás olyan változatával készült, amely számos szokatlan glifet tartalmaz; viszont a szöveg nyelvéről feltételezhető, hogy a készítése korához tartozó nyelvi változatot és hangértékeket alkalmazza. Ennek egy részletét, a 2. ábrán látható hosszabb szót választottuk ki a vizsgálat céljára. A továbbiakban ezt a szót nevezzük feliratnak, vagy általánosan jelszekvenciának. A megfejtendő feliratot jobbról balra írták.



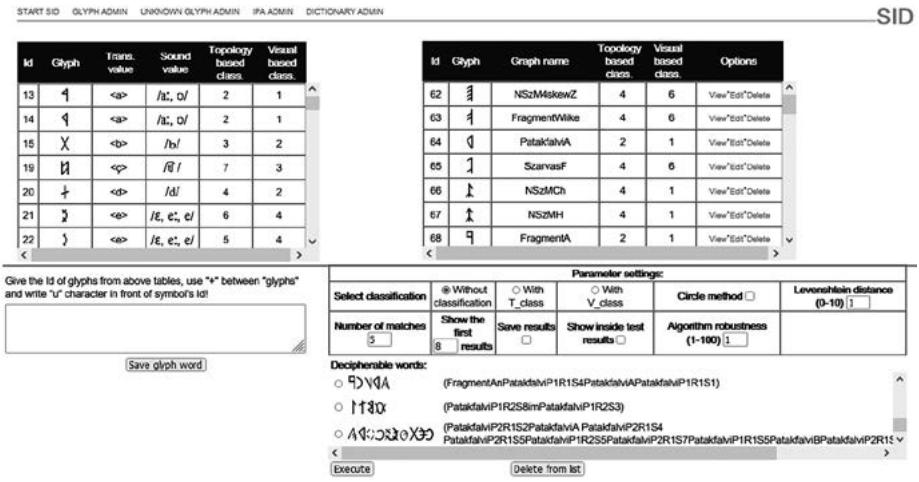
2. ábra: A vizsgált megfejtetlen felirat (jelszekvencia) a Patakfalvi-bibliából (Hosszú 2013b, 251.)

Első lépésként az ismeretlen hangértékűnek tekintett jelszekvenciabeli jelek glifjeit topológiai tulajdonságaikkal együtt feltöltöttük a SID adatbázisába, ezekkel a jelekkel a szöveg (balról-jobbra leírva) a 3. ábra olvasható. Ezen jeleket a SID algoritmus hangértékek vagy átbetűzési értékek nélkül kapta, így a SID számára kezdetben ismeretlenek voltak.

3. ábra: A megfejtetlen felirat digitalizált formája

A SID szoftver PHP/HTML nyelven és MySQL adatbázis használatával készült, a grafikus felhasználói felülete a 4. ábrán látható. A kísérleteket, amelyekhez a futási idők tartoznak, a következő laptop konfiguráción végeztük el: Intel I5-3320M @2.60GHz processzor, 8GB RAM memória, Microsoft Windows 10 operációs rendszer, XAMP vezérlő panel v3.2.2, amely magába foglalja az Apache web- és MySQL adatbázis szervereket.

Több tesztet futtattunk le különböző paraméteropciókkal azért, hogy meghatározzuk a SID szoftver optimális beállításait, így rövidebb feldolgozási idővel pontosabb eredményeket kapjunk. A SID szoftver felhasználói felülete a 4. ábrán látható.



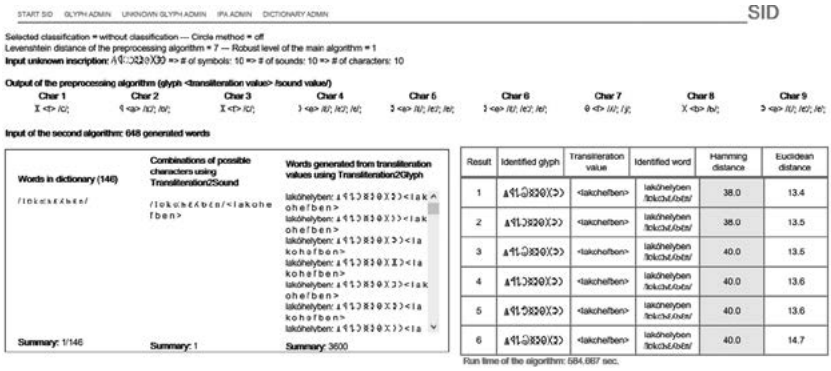
4. ábra: A SID szoftver felhasználói felülete

Az első teszteset során az 1. táblázatban jelölt paramétereket állítottuk be. A *Levenshtein-távolság* paraméter *7*-es értéket kapott ahhoz, hogy találatunk legyen a szótár-adatbázisban. Ennél kisebb érték esetén, a többi paraméter változtatása nélkül nem volt találatunk. A találati pontosság növelésében a *ta-*

lálatszám paraméter nagyobb értékre való állítása segített volna, de hosszabb feliratok esetén – a vizsgált felirat is ezek közé tartozik – az algoritmus futási ideje szignifikánsan megnő, ezért ezen utóbbi paramétert célszerű kis értéken tartani. Az eredményeket az 5. ábra jeleníti meg.

Találatszám	Osztályozás	Körmódszer	Levenshtein-távolság	Erősségszint
1	Osztályozás nélkül	Ki	7	1

1. táblázat: Az algoritmus kezdeti bemeneti paramétereinek beállításai



5. ábra: Az első tesztfuttatás és eredményei

A kezdeti paraméterbeállítások mellett a SID-Előfeldolgozó egyetlen releváns szót talált a szótár-adatbázisban *(lakóhelyben)* a megfejtetlen felirat lehetséges olvasataként. A SID-Fő algoritmus leképezte a *lakóhelyben* szó hangértékeihez tartozó átbetűzési értékeket, majd az átbetűzési értékekhez tartozó glifváltozatok kombinációiból generálta a megfejtendő jelszekvenciához rendelhető összes lehetséges glifszekvenciát. A glifszekvenciákat rendre összehasonlította a megfejtendő jelszekvenciával és kiszámította a topológiai jellemzővektoraik közötti Hamming-távolságot és euklideszi távolságot, továbbá a legrelevánsabb találatokat megjelenítette az alapértelmezett Hamming-távolságok szerint rendezve. Az eredmények meghatározása 584 másodpercet vett igénybe a szoftvernek. A 6. ábrán a megfejtetlen felirat általunk digitalizált formája, míg a 7. ábrán a SID szoftver által az első tesztfuttatás eredményeként meghatározott leghasonlóbb megfejtés látható. A 6. ábra azo-

nos a 3. ábrával, a 7. ábrával való könnyebb összehasonlíthatóság érdekében mutatjuk be ismét.

6. ábra: A megfejtetlen felirat digitalizált formája (azonos a 3. ábrával)

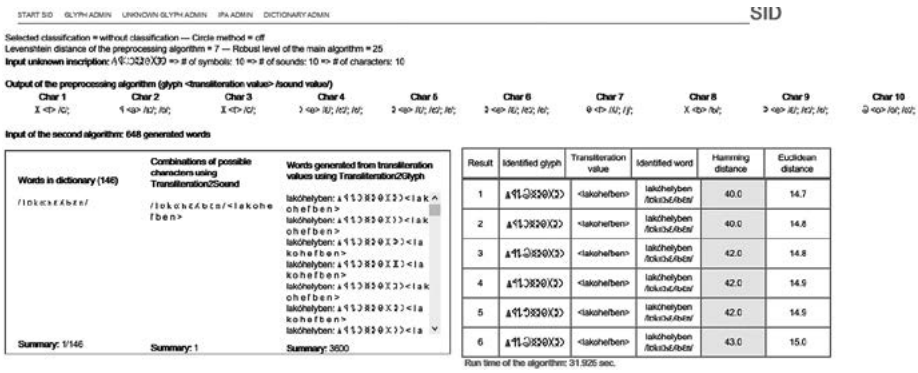
7. ábra: A leghasonlóbb megfejtés az első tesztfuttatás eredményeként

A második teszteset során az algoritmus kezdeti bemeneti paramétereihez képest csak az *erősségszint* paramétert változtattuk *1*-ről *25* értékre (lásd 2. táblázat), ezáltal – ahogy azt az első részben bemutattuk – csak minden 25. generált glifszekvencia-elem került analizálásra a SID-Fő algoritmusban, így a SID-Fő futási sebessége szignifikánsan nőtt a találati pontosság jelentősebb romlása nélkül.

Találatszám	Osztályozás	Körmódszer	Levenshtein-távolság	Erősségszint
1	Osztályozás nélkül	Ki	7	25

2. táblázat: A második tesztfuttatás és eredményei

A második tesztfuttatás eredményeit a 8. ábra mutatja be, ezen látható, hogy a SID-Fő által adott legvalószínűbb megfejtés az előző futtatáshoz hasonlóan maradt a *lakóhelyben* szó, viszont az algoritmus futási idejében jelentős csökkenés látható, hiszen a korábbi 584 másodpercről ~32 másodpercre csökkent.



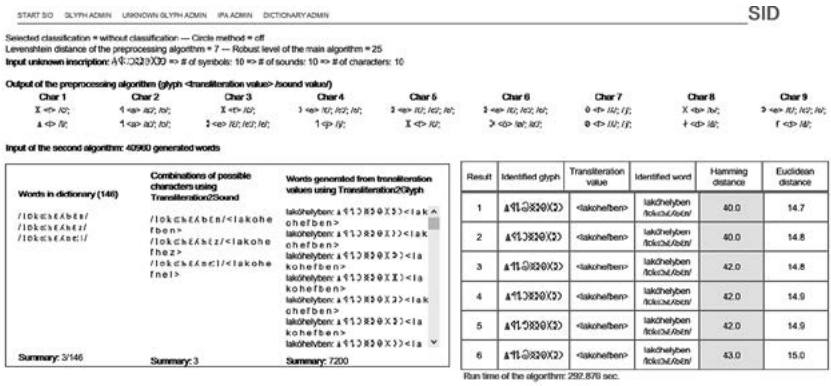
8. ábra: A második tesztfuttatás és eredményei

A második tesztesethez képest a harmadik körben a paramétereknél a *találatszám* paramétert változtattuk *1*-ről *2*-re (lásd 3. táblázat), vagyis a SID-Előfeldolgozó az előző esethez képest több, a megfejteni kívánt felirat jeleihez leghasonlóbb glifet vizsgált. Az ezekből a glifekből kombinációkkal képzett glifszekvenciákhoz tartozó hangértékszekvenciák és a szótárban levő szóhalmaz metszetének eredményeképp több újabb találattal tért vissza lehetséges megfejtések gyanánt, ezek azután a SID-Fő algoritmus bemenetét képezték.

Találatszám	Osztályozás	Körmódszer	Levenshtein-távolság	Erősségszint
2	Osztályozás nélkül	Ki	7	25

3. táblázat: A harmadik tesztfuttatás és eredményei

Az eredményeket a 9. ábra tartalmazza. Megfigyelhető, hogy a *lakóhelyben* szó mellé újabb két találat került ki a szótárból a *lakóhelynél* és a *lakóhelyhez* szavak. Mindezek ellenére a SID-Fő algoritmus legrelevánsabbként – mint leghasonlóbb találatot az eredeti felirathoz – a *lakóhelyben* szót hozta ki eredményül. A szoftver futási ideje jelentősen növekedett az előző tesztesethez képest, mivel a SID-Fő algoritmus a korábbi egy találat helyett három találattal dolgozott.

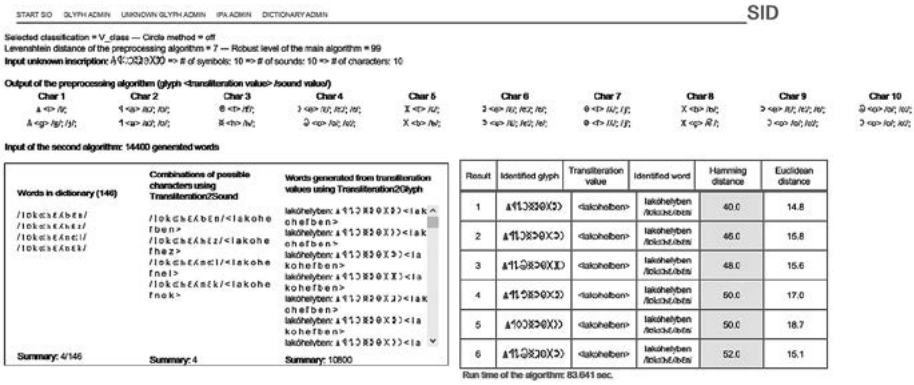


9. ábra: A harmadik tesztfuttatás és eredményei

Az utolsó tesztesetnél két algoritmusgyorsításért felelős paramétert módosítottunk, az erősségszint paramétert *99*-re, míg az o*sztályozás* paramétert *V-osztályozóra* állítottuk (lásd 4. táblázat). Ez utóbbi paraméter a gyorsítás mellett relevánsabb találatokat is eredményezhet a SID-Előfeldolgozó kimenetén, ahogy ezt a példa is szemlélteti. Az eredmények a 10. ábrán láthatóak. Az előző három szótáradatbázisbeli találat a "*V-osztályozónak*" köszönhetően kiegészült a "*lakóhelynek*" szóval, de ennek ellenére a legrelevánsabb megoldás még mindig a "*lakóhelyben*" szó lett. Az *erősségszint* paraméter további növelésével az algoritmus futási ideje csökkent, annak ellenére, hogy három szó helyett négy került a SID-Fő bemenetére.

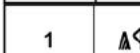
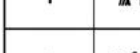
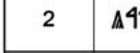
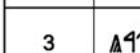
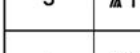
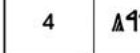
Találatszám	Osztályozás	Körmódszer	Levenshtein-távolság	Erősségszint
2	V-osztályozó	Ki	7	99

4. táblázat: A negyedik tesztfuttatás és eredményei



10. ábra: A negyedik tesztfuttatás és eredményei

Itt szeretnénk megjegyezni, hogy a negyedik tesztfuttatás során beállított paraméterek mellett az eredmények rangsorában a 5. helyen áll a megfejtés azon glifábrázolása, amelyben a székely-magyar rovás <k> rombusz alakú glifje is helyet kapott (lásd 11. ábra). Ez azért fontos felismerés, mivel a megfejtendő szó szemrevételezése során a benne ábrázolt <k> graféma glifreprezentációja vizuálisan egy rombusz alakú glifre hasonlít jobban, viszont a szigorúan vett topológiai tulajdonságai (nyitott, az egyes vonalai nem érintkeznek egymással, a szakaszok végpontjainak száma, kereszteződések száma stb.) figyelembe vétele alapján a SID-algoritmus az általa számított első helyen szereplő találatában a <k> grafémára a nyitott alakú glifet hozta ki.

Result	Identified glyph	Transliteration value	Identified word	Hamming distance	Euclidean distance
1		<lakoheľben>	lakóhelyben /lɒkoːhɛʎbɛn/	40.0	14.8
2		<lakoheľben>	lakóhelyben /lɒkoːhɛʎbɛn/	46.0	15.8
3		<lakoheľben>	lakóhelyben /lɒkoːhɛʎbɛn/	48.0	15.6
4		<lakoheľben>	lakóhelyben /lɒkoːhɛʎbɛn/	50.0	17.0
5		<lakoheľben>	lakóhelyben /lɒkoːhɛʎbɛn/	50.0	18.7
6		<lakoheľben>	lakóhelyben /lɒkoːhɛʎbɛn/	52.0	15.1

11. ábra: A negyedik tesztfuttatás eredményei

A SID-algoritmus az ismeretlen jelentésű jelszekvenciák megfejtésében olyan eszközt adhat a kutatók kezébe, amely segíthet támpontot adni egy jelszekvencia megfejtése során. Ugyanakkor a megbízható működés érdekében egy vizsgált jelszekvenciával kapcsolatban további metaadatok adatbázisba történő felvétele is szükségessé válhat.

Következtetések

A régi feliratokban (általánosan jelszekvenciákban) gyakran előfordulnak hibásan írt jelek, hiányos szavak vagy különleges, régebbi korokban használt glifváltozatok, melyek megnehezítik a megfejtésüket. Jelen cikkben bemutattunk egy SID nevű összetett eljárást, amely a régi feliratok jelentését azonosítja távolságmetrikák és vektorműveletek használatával az ismert glifek és a megfejtetlen feliratbeli jelek geometriai-topológiai tulajdonságai alapján. Ezen jelek távolságmetrikákkal, osztályozási és algoritmus-gyorsítási módszerekkel kerülnek feldolgozásra.

A rendszer teljesítményét egy valós megfejtési eseten mutattuk be, ahol egy a SID algoritmusa számára ismeretlen jelentésű, egyszavas felirat jelentését si-

került megfejteni. Az eddigi vizsgálataink azt mutatják, hogy a topológiai jellegek helyes meghatározásával és a megfelelő távolságmetrikák alkalmazásával olyan jelentésazonosító algoritmus készíthető, amely figyelembe veszi, hogy egy írás (általánosan mintarendszer) fejlődése során feledésbe merülhettek a hozzá tartozó grafémák (általánosan szimbólumok) egyes glifváltozatai.

A cikk egy SID nevű összetett eljárást ír le a megfejtetlen feliratok jelentésének azonosítására a szimbólumok és a betűk topológiai jellemzőin alapuló távolságmetrikák és vektorműveletek segítségével. A SID algoritmust megvalósító szoftver különböző módszereket alkalmaz az algoritmus gyorsítására és a feldolgozási idő csökkentésére, valamint az algoritmus hatékonyságának javítására és az ismeretlen jelentésű feliratok megfejtési valószínűségének növelésére. Az algoritmusok és módszerek rövid bemutatása után a rendszer teljesítményét egy valós megfejtési eseten mutattuk be. A megfejtetlen jelszekvenciát a SID algoritmus sikeresen megfejtette. A SID alapalgoritmusát kiegészítő, gyorsító célzatú módszerek hatását több tesztfuttatással, különböző paraméterbeállításokkal mutattuk be.

A SID-algoritmus felhasználja a glifek és a megfejtetlen jelek tipikus topológiai jellemzőit, ezek képezik a kutatás alapját. Megállapítottuk, hogy a topológiai jellemzők helyes meghatározásával és kiválasztásával, valamint a megfelelő távolságmetrikával kombinálva hatékony megfejtési eredmények érhetők el alacsony hardvererőforrás-felhasználás mellett.

Jelen tanulmány a székely-magyar rovás írással készült feliratok megfejtésére koncentrál, de a bemutatott módszer kiterjeszthető más írásokra, sőt általánosan mintarendszerekre is. A SID megvalósítása segítséget nyújthat a régészek és a paleográfusok számára a régi feliratok megfejtésében.

HIVATKOZOTT IRODALOM

Amato et al. 2016. Giuseppe Amato – Fabrizio Falchi – Lucia Vadicamo: Visual Recognition of Ancient Inscriptions Using Convolutional Neural Network and Fisher Vector. *ACM Journal on Computing and Cultural Heritage* 9. (2016) 4. sz. article no. 21., 1–24.

Bag et al. 2011. S. Bag – G. Harit – P. Bhowmick: Topological features for recognizing printed and handwritten Bangla characters. In: *MOCR_AND '11: Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data,* September 2011. ACM, New York, 2011. Article no. 10., 1–7.

Barmpoutis et al. 2010. A. Barmpoutis – E. Bozia – R. S. Wagman: A novel framework for 3D reconstruction and analysis of ancient inscriptions. *Machine Vision and Applications* 21. (2010) 6. sz. 989–998.

Basson – Davel 2013. W. D. Basson – M. H. Davel: Category-based phoneme -to-grapheme transliteration. In: F. Bimbot (ed.): *Conference: Interspeech*, Lyon, France, August 2013. International Speech Communication Association, 2013. 1956–1960.

Chaudhuri et al 2017. A. Chaudhuri – K. Mandaviya – P. Badelia – S. K. Ghosh: *Optical Character Recognition Systems for Different Languages with Soft Computing.* Springer, New York, 2017.

Chen et al. 2004. D. Chen – J.-M. Odobez – H. Bourlard: Text detection and recognition in images and video frames. *Pattern Recognition* 37. (2004) 595–608.

Cunderlik–Burn 2016. J. M. Cunderlik – D. H. Burn: Switching the pooling similarity distances: Mahalanobis for Euclidean. *Water Resources Research* 42. (2016) W03409. sz.

Daggumati–Revesz 2019. S. Daggumati – P. Revesz: Data mining ancient scripts to investigate their relationships and origins. In: *Conference: the 23rd International Database Applications & Engineering Symposium*, June 2019. Association for Computing Machinery, New York, 2019. Article no. 26. 1–10.

Das et al. 2012. N. Das – J. M. Reddy – R. Sarkar – S. Basu – M. Kundu – M. Nasipuri – D. K. Basu: A Statistical-Topological Feature Combination for Recognition of Handwritten Numerals. *Applied Soft Computing* 12. (2012) 8. sz. 2486–2495.

Gósy 2004. Gósy Mária: *Fonetika, a beszéd tudománya.* Osiris Kiadó, Budapest, 2004.

Hamming 1950. Richard W. Hamming: Error detecting and error correcting codes. *The Bell System Technical Journal* 29 (1950) 2. sz. 147–160.

Hosszú 2013a. Gábor Hosszú: The Rovas: A Special Script Family of the Central and Eastern European Languages. *Acta Philologica* 44. (2013) 91–102.

Hosszú 2013b. Gábor Hosszú: *Heritage of Scribes. The Relation of Rovas Scripts to Eurasian Writing Systems,* 3rd, revised and extended edition. Rovas Foundation, Budapest, 2013. http://google.hu/books?id=TyK8azCqC34C&pg

Hosszú 2014a. Hosszú Gábor: Topológiai eltérések minimalizálására visszavezetett graféma leszármazási vizsgálatok. In: Cserny László – Hadaricsné Dudás Nóra – Nagy Bálint (szerk.): *Informatika Korszerű Technikái Konferencia* (2012. november 16–17.). Dunaújvárosi Főiskola – Új Mandátum Könyvkiadó, Budapest, 2014. 60–71.

Hosszú 2014b. Gábor Hosszú: Mathematical Statistical Examinations on Script Relics. In: V. Bhatnagar (ed.): *Data Mining and Analysis in the Engineering Field.* Chapter 8. Information Science Reference, Hershey (PA) – New York, 2014. 142–158.

Hosszú 2017. Gábor Hosszú: Phenetic Approach to Script Evolution. In: Hannah Busch, Franz Fischer, Patrick Sahle (eds.), *Kodikologie und Paläographie im digitalen Zeitalter 4 – Codicology and Palaeography in the Digital Age 4*. Schriften des Instituts für Dokumentologie und Editorik 11. Books on Demand, Norderstedt, 2017. 179–252.

Hosszú 2019. Hosszú Gábor: Írásemlékek grafémaalakjainak térstatisztikai és fenetikai elemzése. In: Zelliger Erzsébet (szerk.): *Rovás – magyar nyelvtörténet – művelődéstörténet.* Magyarságkutató Intézet, Budapest, 2019. 120–450.

Hosszú 2021. Gábor Hosszú: *Scriptinformatics. Extended Phenetic Approach to Script Evolution.* Nap Kiadó, Budapest, 2021. http://napkiado.hu/media/Hosszu_Gabor-Scriptinformatics.pdf

Hosszú–Kovács 2016. Gábor Hosszú – Ferenc Kovács: Topological analysis of ancient glyphs. In: *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC),* October 9–12, 2016. IEEE, Budapest, 2016. 2248–2253.

Jäger 2019. G. Jäger: Computational historical linguistics. *Theoretical Linguistics* 45. (2019) 3–4. sz. 151–182.

Kanimozhi 2012. J. K. Kanimozhi: Skeletal Graph Based Topological Feature Extraction of an Object. *Journal of Computer Applications* 5. (2012) EICA2012-1. sz. 111–118.

Levenshtein 1966. *Vladimir I. Levenshtein: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10. (1966) 8. sz. 707–710.*

Lin et al. 2014. Y.-S. Lin – J.-Y. Jiang – S.-J. Lee: A Similarity Measure for Text Classification and Clustering. *IEEE Transactions on Knowledge and Data Engineering* 26. (2014) 7. sz. 1575–1590.

Nguyen et al. 2016. Thanh Phuong Nguyen – Antoine Manzanera – Walter G. Kropatsch – Xuan Son Nguyen: Topological Attribute Patterns for texture recognition. *Pattern Recognition Letters* 80. (2016) 91–97.

Pardede et al. 2012. Raymond Eliza Ivan Pardede – Loránd Lehel Tóth – Gábor Hosszú – Ferenc Kovács: Glyph Identification Based on Topological Analysis. In: *Proceedings of the PhD Workshops at BME,* March 9, 2012. BME, Budapest, 2012. 99–103.

Pardede et al. 2016. Raymond Eliza Ivan Pardede – Loránd Lehel Tóth – György András Jeney – Ferenc Kovács – Gábor Hosszú 2016. Four-Layer Grapheme Model for Computational Paleography. *Journal of Information Technology Research (JITR)* 9. (2016) 4. sz. 64–82.

Putra–Supriana 2015. M. E. W. Putra – I. Supriana: Structural Offline Handwriting Character Recognition Using Levenshtein Distance. In: *5th International Conference on Electrical Engineering and Informatics,* August 2015. IEEE, Piscataway (NJ), 2015. 35–40.

Rahman et al. 2019. M. Rahman – S. Islam – R. S. Aktaruzzaman: Convolutional neural networks performance comparison for handwritten Bengali numerals recognition. *SN Applied Sciences* 1. (2019) article 1660.

Rama et al. 2018. T. Rama – J.-M. List – J. Wahle – G. Jäger: Are automatic methods for cognate detection good enough for phylogenetic reconstruction in historical linguistics? In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, June 2018, vol. 2 (Short Papers). Association for Computation Linguistics, New Orleans, 2018. 393–400.

Rousopoulos et al. 2011. P. Rousopoulos – M. Panagopoulos – C. Papaodysseus – F. Panopoulou – D. Arabadjis – S. Tracy – F. Giannopoulos – S. Zannos: A new approach for ancient inscriptions' writer identification. In: *17th International Conference on Digital Signal Processing (DSP)*. IEEE, Piscataway (NJ), 2011. 1–6.

Sanchez et al. 2013. Jorge Sanchez – Florent Perronnin – Thomas Mensink – Jakob Verbeek: Image Classification with the Fisher Vector: Theory and Practice. *International Journal of Computer Vision* 105. (2013) 3. sz. 222–245.

Sapirstein 2019. P. Sapirstein: Segmentation, Reconstruction, and Visualization of Ancient Inscriptions in 2.5D. *Journal on Computing and Cultural Heritage* 12. (2019) 2. sz. article no. 15.

Shehu et al. 2015. G. S. Shehu – A. M. Ashir – A. Eleyan: Character recognition using correlation & hamming distance. In: *23nd Signal Processing and Communications Applications Conference (SIU)*. IEEE, Piscataway (NJ), 2015. 755–758.

Snyder et al. 2010. B. Snyder – R. Barzilay – K. Knight: A Statistical Model for Lost Language Decipherment. In: *Conference: ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* Uppsala, Sweden, July 2010. 1048–1057.

Stauffer et al. 2019. M. Stauffer – P. Maergner – A. Fischer – R. Ingold – K. Riesen: Offline Signature Verification using Structural Dynamic Time Warping. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, Los Alamitos (CA), September 2019.

Tirandaz és mtsai. 2017. H. Tirandaz – M. Ahmadnia – H. Tavakoli: Geometric-Topological Based Arabic Character Recognition. *A New Approach, Journal of Theoretical and Applied Information Technology* 95. (2017) 15. sz. 3692–3702.

Tóth et al. 2015. Loránd Lehel Tóth – Raymond Eliza Ivan Pardede – Gábor Hosszú: Novel Algorithmic Approach to Deciphering Rovash Inscriptions. In: M. Khosrow-Pour (Ed.): *Encyclopedia of Information Science and Technology* Information Science Reference, Hershey (PA), 2015[3]. 7222–7233.

Tóth et al. 2016a. Loránd Lehel Tóth – Raymond Eliza Ivan Pardede – György András Jeney – Ferenc Kovács – Gábor Hosszú: Application of the Cluster Analysis in Computational Paleography. In: Pijush Samui (Ed.): *Handbook of Research on Advanced Computational Techniques for Simulation-Based Engineering.* Engineering Science Reference, Hershey (PA), 2016. 525–543.

Tóth et al. 2016b. Loránd Lehel Tóth – Raymond Eliza Ivan Pardede – György András Jeney – Ferenc Kovács – Gábor Hosszú: Preprocessing Algorithm for Deciphering Historical Inscriptions Using String Metric. *International Journal of Engineering and Technology Innovation (IJETI)* 6. (2016) 3. sz. 202–213.

Tóth et al. 2021. Loránd Lehel Tóth – Ferenc Kovács – Gábor Hosszú: Deciphering Historical Inscriptions Using Machine Learning Methods. In: Shifeng Liu – Gábor Bohács – Xianliang Shi – Xiaopu Shang – Anqiang Huang (eds.): *LISS 2020. Proceedings of the 10th International Conference on Logistics, Informatics and Service Sciences.* Springer, Singapore, 2021. 419–435. https://doi.org/10.1007/978-981-33-4359-7_30

Tóth–Hosszú 2019. Loránd Lehel Tóth – Gábor Hosszú: A New Topological Method for Examining Historical Inscriptions. *Journal of Information Technology Research* 12. (2019) 2. sz. 1–16.

Zaghloul et al 2011. R. I. Zaghloul – E. F. AlRawashdeh – D. M. K. Bader: Multilevel Classifier in Recognition of Handwritten Arabic Characters. *Journal of Computer Science* 7. (2011) 4. sz. 512–518.

Zhao–Sahni 2019. C. Zhao – S. Sahni: String correction using the Damerau-Levenshtein distance, *BMC Bioinformatics* 20. (2019) article no. 277.

Meaning identification algorithm based on pattern evolution

ABSTRACT: Determining the meaning of undeciphered inscriptions (more generally, graph sequences) made with scripts (more generally, pattern systems) involves finding the symbol sequence implemented by the graph sequence. The SID meaning identification procedure finds the symbol sequence that best represents the meaning of the graph sequence using distance metrics and vector operations. It does this by searching for the corresponding symbol sequence from a dictionary database based on glyphs of symbols in a given pattern system and topological features of the graphs in the graph sequence under investigation. The dictionary database corresponds to the age at which the graph sequence was created and the hypothetical language of the symbol sequence realized by the graph sequence. The procedure takes into account the fact that some glyphs associated with symbols of a pattern system evolving over time may have been forgotten during the evolution of the pattern system. The performance of the SID algorithm has been demonstrated on a real deciphering case.

KEYWORDS: computational palaeography, distance metric, meaning identification, pattern evolution, pattern system, scriptinformatics, Székely-Hungarian Rovash, topological attribute