



A nyelvtechnológia alapjai 4.

Unifikációs morfológia,
szófaji egyértelműsítés,
alkalmazások

2019/2020. tanév, I. félév

Gráfok, fák, DAG-ok

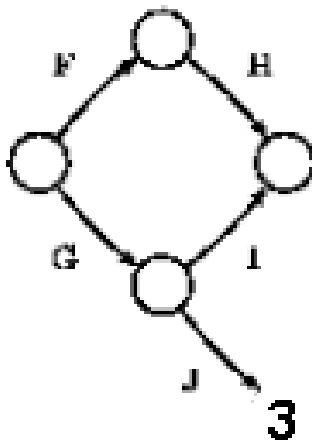
- **Gráf:** egy olyan objektumhalmaz reprezentációja, ahol egyes objektumpárokat kapcsolatok kötnek össze; az objektumokat csomópontoknak, az összekötéseket éleknek mondjuk
- **Ösvény:** egy adott u_0 csomópontból kiinduló, u_n -be vezető olyan élsorozat, melyben minden él u_{i-1} -ből u_i -be ($i=1, \dots, n-1$) vezető él végpontjában levő csomópontból kiindul egy u_{i+1} -be vezető él
- u és v csomópontok **kapcsoltak**, ha a gráfnak van u -ból v -be menő ösvénye, és a gráf **összefüggő** gráf, ha minden különböző csomópontja kapcsolt
- **Írányított gráf:** csomópontok halmaza élek rendezett párjaival
- **Gyökér:** speciális csomópont, amelyből bármely csomóponthoz vezet ösvény
- **Levél:** kimenő él nélküli csomópont
- **Írányított körmentes gráf (DAG):** irányított gráf, kör nélkül

Az unifikáció

- A jegyszerkezetek **irányított körmentes gráfokkal** reprezentálhatók, ahol a csomópontok felelnek meg a változóknak és az ösvények a változóneveknek
- A jegyszerkezeteket leggyakrabban jegy-érték **mátrixokként** írják le, melyeknek két oszlopuk van (az egyik a jegyek nevének, a másik az értékeinek)
- Az **érték** vagy atomi érték vagy egy másik jegyszerkezet
- **Jegyszerkezet-unifikáció:** két jegyszerkezeten működő művelet, mely csak akkor hiúsul meg, ha van olyan ösvény mindkét szerkezetben, melyeknek eltérő atomi értékei vannak; minden más esetben a létrejövő szerkezet az eredeti két szerkezet összes ösvényét tartalmazza (az azonosakat egybeejtve)
- Az unifikáció absztrakt művelet, és **NEM** a morfológiához tartozik, csak a legegyszerűbb formáját itt tárgyaljuk először

Jegyszerkezetek mint DAG-ok

A jegyszerkezet egy gráf:



A jegyszerkezet mint attribútum-érték mátrix:

$$\left[\begin{array}{c|c} F & H & \boxed{1} \\ \hline G & \begin{bmatrix} I & \boxed{1} \\ J & 3 \end{bmatrix} \end{array} \right]$$

•Source: Frederik Fouvry - <http://www.dfki.de/compling/pdfs/ubg-slides.pdf>

Az unifikáció művelete

Az unifikáció olyan művelet,
mely egyedül akkor nem hajtható végre,
ha a tagokban egyazon jegy
különböző értékekkel szerepel:

[NUMBER SG] U [NUMBER PL] = *fail!*

de:

[NUMBER SG] U [NUMBER SG] = [NUMBER SG]

[NUMBER SG] U [NUMBER []] = [NUMBER SG]

[NUMBER SG] U [PERSON 3] = [[NUMBER SG][PERSON 3]]

Példák unifikációra

$$\begin{bmatrix} \text{CATEGORY} & \textit{noun} \end{bmatrix} \sqcup \begin{bmatrix} \text{NUMBER} & \textit{singular} \end{bmatrix} = \begin{bmatrix} \text{CATEGORY} & \textit{noun} \\ \text{NUMBER} & \textit{singular} \end{bmatrix}$$

$$\begin{bmatrix} \text{CAT} & \square \end{bmatrix} \sqcup \begin{bmatrix} \text{CAT} | \text{CASE} & \textit{accusative} \end{bmatrix} = \begin{bmatrix} \text{CAT} | \text{CASE} & \textit{accusative} \end{bmatrix}$$

$$\begin{bmatrix} \text{F} & \boxed{1} \\ \text{H} & \boxed{1} \end{bmatrix} \sqcup \begin{bmatrix} \text{F} & \square \\ \text{H} | \text{G} & \square \end{bmatrix} = \begin{bmatrix} \text{F} & \boxed{1} \begin{bmatrix} \text{G} & \square \end{bmatrix} \\ \text{H} & \boxed{1} \end{bmatrix}$$

$$\begin{bmatrix} \text{CATEGORY} & \textit{noun} \end{bmatrix} \sqcup \begin{bmatrix} \text{CATEGORY} & \textit{verb} \end{bmatrix} = \text{fail}$$

$$\begin{bmatrix} \text{AGR} & \boxed{1} \begin{bmatrix} \text{NUM} & \textit{sg} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR:} & \boxed{1} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & \textit{third} \end{bmatrix} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \text{AGR} & \boxed{1} \begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{PERS} & \textit{third} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

•Shieber, S.. An introduction to unification-based approaches to grammar . CSLI Lecture Notes 15. Stanford (1986)

Az unifikáció implementációkban

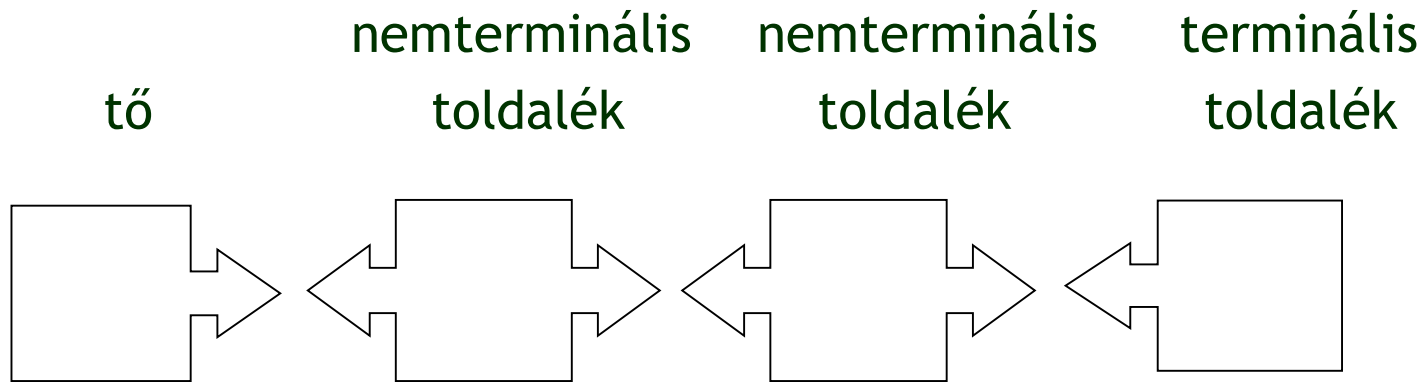
Destruktív unifikáció:

- Két szerkezet unifikációjakor az egyik megszűnik, és helyén létrejön az eredményszerkezet
- Gyors
- Bizonyos esetekben nem kívánt hatása is lehet (pl. ha egy szabályt alkalmazunk, annak formáját nem szeretnénk a művelet következtében megváltoztatni)

Nem-destruktív unifikáció (unifikálhatóság-ellenőrzés):

- A kiinduló szerkezetek nem változnak, az eredmény egy új, harmadik szerkezet
- Az implementációkban igen gyakori
- Könnyű kezelni, viszont másolási műveletet igényel
- Az unifikálhatóság-ellenőrzés egy reláció, igen/nem eredménnyel, más kimenet nélkül

Szóalaktani alapséma



(relatív) tő / relatív toldalék

relatív tő / relatív toldalék

relatív tő / (relatív) toldalék

A HUMOR morfológiai rendszer

- ❑ High-speed Unification Morphology
- ❑ Az unifikáció mint művelet NEM a morfológia sajátja (lesz ilyen a többi nyelvi szinteken is), csak itt tárgyaljuk először
- ❑ Unifikáció van a logikai programozásban is, csak ott az argumentumok sorrendje is rögzített (Prolog)
- ❑ A lényeg: jegy-érték párok
- ❑ Az unifikáció destruktív művelet, de a HUMOR esetében csak unifikálhatósági relációról beszélünk
- ❑ Minden tulajdonság jegyként jelenik meg (az is, ami literálisan is jól látszik!)
- ❑ A folytatási osztályok találkozási pontjain működik
- ❑ Nincs más aktív művelet, csak az unifikálhatóság-ellenőrzés

Bináris kérdések a magyar morfo-fonológiáról

		$\alpha = +$	$\alpha = -$
1	α névszó	névszó	ige
2	α fn	főnév	melléknév, számnév
3	α szótári	szótári alapalak	nem szótári alapalak
4	α elöl	elöl képzett	hátsó képzett
5	α kerek	ajakkerekítéses	nem ajakkerekítéses
6	α PL	többes szám	nem állhat többes számban
7	α PLkötő	PL kötőhanggal	PL nem kötőhanggal
8	α PERS	birt. szem.ragos	nem kap birt. szem.ragot
9	α ACC	van tárgyesete	nem tárgyesetes
10	α ACCkötő	ACC kötőhanggal	ACC nem kötőhanggal
11	α DAT	van részesesete	nincs részesesete
12	α INS: β	van eszk.h.esete	nincs eszk.h.esete
13	α ÁS	-ás/-és képzős	nem kap -ás/-és képzőt
14	...		

Szótövek tára

hó []
[+névszó +fn +szótári -elől -kerek -PL
-PERS -ACC -ACCKötő +DAT +INS:V]

hav []
[+névszó +fn -szótári -elől -kerek +PL
+PLkötő +PERS +ACC -DAT -INS]

képez []
[-névszó +szótári +elől -kerek -ÁS]

képz []
[-névszó -szótári +elől -kerek +ÁS]

Toldalékok tára

<i>ás</i>	$[-\text{névszó} -\text{elöl} +\text{ÁS}]$ $[+\text{névszó} +\text{fn} +\text{szótári} -\text{elöl} -\text{kerek} +\text{PL}$ $+ \text{PLkötő} +\text{ACC} -\text{ACCkötő} +\text{DAT} +\text{INS:S}]$
<i>és</i>	$[-\text{névszó} +\text{elöl} +\text{ÁS}]$ $[+\text{névszó} +\text{fn} +\text{szótári} +\text{elöl} -\text{kerek} +\text{PL}$ $+ \text{PLkötő} +\text{ACC} -\text{ACCkötő} +\text{DAT} +\text{INS:S}]$
<i>ak</i>	$[+\text{névszó} -\text{elöl} -\text{kerek} +\text{PL} +\text{PLkötő}]$ $[+\text{névszó} -\text{elöl} -\text{kerek} -\text{PL} -\text{PERS} +\text{ACC}$ $+ \text{ACCkötő} +\text{DAT} +\text{INS:K}]$
<i>ek</i>	$[+\text{névszó} +\text{elöl} -\text{kerek} +\text{PL} +\text{PLkötő}]$ $[+\text{névszó} +\text{elöl} -\text{kerek} -\text{PL} -\text{PERS} +\text{ACC}$ $+ \text{ACCkötő} +\text{DAT} +\text{INS:K}]$
<i>nak</i>	$[+\text{névszó} -\text{elöl} +\text{DAT}]$ $[]$
<i>nek</i>	$[+\text{névszó} +\text{elöl} +\text{DAT}]$ $[]$

Az unifikáció pseudokódos definíciója

function UNIFY(*f1*, *f2*) **returns** *fstructure* or failure

f1-real $\leftarrow$ Real contents of *f1*

f2-real $\leftarrow$ Real contents of *f2*

if *f1-real* is null **then**

f1.pointer $\leftarrow$ *f2*

return *f2*

else if *f2-real* is null **then**

f2.pointer $\leftarrow$ *f1*

return *f1*

else if *f1-real* and *f2-real* are identical **then**

f1.pointer $\leftarrow$ *f2*

return *f2*

else if both *f1-real* and *f2-real* are complex feature structures **then**

f2.pointer $\leftarrow$ *f1*

for each *feature* **in** *f2-real* **do**

other-feature $\leftarrow$ Find or create

 a feature corresponding to *feature* in *f1-real*

if UNIFY(*feature.value*, *other-feature.value*) **returns** failure **then**

return failure

return *f1*

else return failure



Az unifikációs morfológia működése

<i>hó</i>	[+névszó +fn +szótári -elöl -kerek -PL -PERS -ACC -ACCKötő +DAT +INS:V]
<i>*hav</i>	[+névszó +fn <u>-szótári</u> -elöl -kerek +PL +PLkötő +PERS +ACC +DAT -INS]
<i>hó+nak</i>	[+névszó +fn +szótári -elöl -kerek -PL -PERS -ACC -ACCKötő +DAT +INS:V] [+névszó -elöl +DAT]
<i>*hav+nak</i>	[+névszó +fn -szótári -elöl -kerek +PL +PLkötő +PERS +ACC <u>-DAT</u>] [+névszó -elöl <u>+DAT</u>]
<i>hav+at</i>	[+névszó +fn -szótári -elöl -kerek +PL +PLkötő +PERS +ACC -DAT] [+névszó -elöl +ACC]
<i>*hó+vel</i>	[+névszó +fn +szótári <u>-elöl</u> -kerek -PL -PERS -ACC -ACCKötő +DAT +INS:V] [+névszó <u>+elöl</u> +INS:V]
<i>képz+és+nek</i>	[-névszó -szótári +elöl -kerek +ÁS] [-névszó +elöl +ÁS] [+névszó +fn +szótári +elöl -kerek +PL +PLkötő +ACC -ACCKötő +DAT +INS:S] [+névszó +elöl +DAT]

Az unifikációról általában

- ☐ Az unifikáció mint művelet **NEM** a morfológiához tartozik, mindössze egy egyszerűbb változatát itt tárgyaljuk először
- ☐ Az unifikációt elsősorban irányított ciklusmentes gráfok (directed acyclic graph, DAG) esetében szokás használni
- ☐ A morfológiai leírás esetében egyszerűbb struktúrákon működtetjük az unifikációt
- ☐ A DAG több mint fa, és épp ezáltal használhatóbb nyelvi szerkezetek leírására, hogy egy csomópontnak a nyelvi szerkezetekben sokszor kell, hogy több szülője lehessen
- ☐ A közös csomópontok elsősorban a szintaktikai szerkezetek kezelésénél válnak fontossá

Morfo-fonológiai „guesser”

kacsónak	✓ N
kacsóna + k	* N + PL
kacsón + ak	* N + PL
kacsó + nak	* V + PL3
kacsó + nak	✓ N + DAT
kacs + ó + nak	* V + PART + DAT
ka csónak	* N N

Innentől a szóalak-gyakoriság dönt:

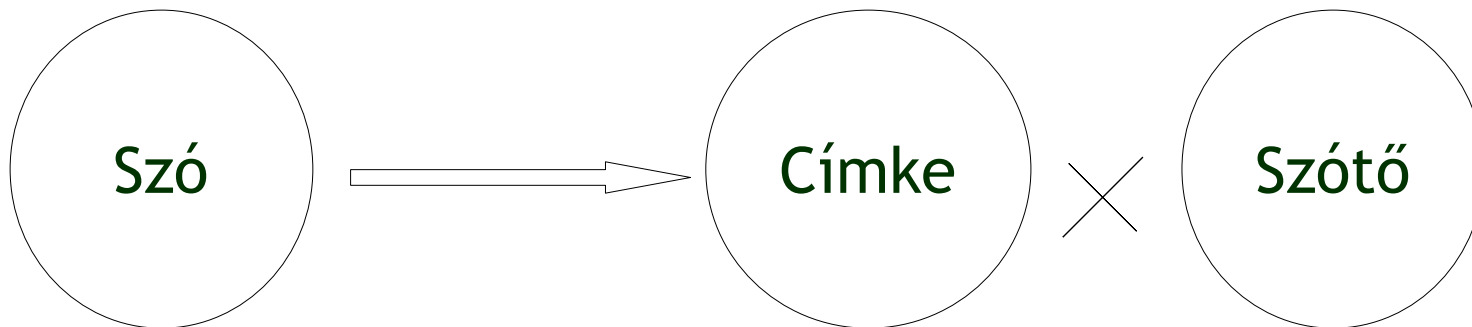
- kacsónak, *kacsónakot, *kacsónakra, ... → {}
- kacsó, kacsót, kacsóra, kacsói, ... → {kacsó}



Szófaji egyértelműsítés

A szófaji egyértelműsítési feladat

Egyértelműsítő:



{tokenizált mondat, címkekészlet}
→ **szófaji egyértelműsítő** →
{címkézett (és szótövezett) mondat}

Markov-modellek

- ☐ **Markov-lánc:** diszkrét sztochasztikus folyamatot ír le
- ☐ **Markov-folyamat:** adott jelenbeli állapot mellett a rendszer jövőbeni állapota nem függ a múltbeliektől
- ☐ **Rejtett Markov-modell (HMM):** csak a modell működésének az eredményét (a kimenetet, azaz a generált szekvenciát) ismerhetjük, a modell maga és a paraméterei számunkra ismeretlenek
- ☐ Következésképp csak a kimenetből következtethetünk a modell felépítésére és a működését leíró paraméterekre (az átmeneti és a kibocsátási valószínűségekre)
- ☐ A modell karakterisztikus tulajdonsága tehát a rejtett állapot, azaz a (sztochasztikus) állapotfejlődés és annak passzív (sztochasztikus) megfigyelése

Az alapvető szófaji egyértelműsítő módszerek

- ❑ Szabály-alapú módszerek
 - ❑ Megszorításos modellek
 - ❑ Transzformációs modellek
- ❑ Valószínűségi módszerek (W: szósorozat, T: címkesorozat)
 - ❑ Rejtett Markov-modellen alapuló módszerek → a legjobb címkesorozat, ami maximalizálja a $P(T|W)$ -t:

$$\begin{aligned}\hat{T} &= \operatorname{argmax}_T P(T|W) \\ &= \operatorname{argmax}_T P(W|T)P(T) \\ &= \operatorname{argmax}_T \prod_i P(\text{word}_i|\text{tag}_i) \prod_i P(\text{tag}_i|\text{tag}_{i-1})\end{aligned}$$

- ❑ Maximum entrópián alapuló módszerek → a $P(T|W)$ kiszámítása a lehetséges címkesorozatok közül választva:

$$\begin{aligned}\hat{T} &= \operatorname{argmax}_T P(T|W) \\ &= \operatorname{argmax}_T \prod_i P(t_i|w_i, t_{i-1})\end{aligned}$$

Egy megszorításos egyértelműsítő: EngCG

- ❑ Kb. 3600 szabály (az angolra)
- ❑ Pl. a (*That round*) *table might collapse*. mondat (vége):
 - "table" N NOM SG
 - "table" <SVO> V SUBJUNCTIVE VFIN @+FMAINV
 - "table" <SVO> V IMP VFIN @+FMAINV
 - "table" <SVO> V INF
 - "table" <SVO> V PRES -SG3 VFIN @+FMAINV
 - "might" <-Indef> N NOM SG
 - "might" V AUXMOD VFIN @+FAUXV
 - "collapse" N NOM SG
 - "collapse" <SV> <SVO> V IMP VFIN @+FMAINV
 - "collapse" <SV> <SVO> V INF
 - "collapse" <SV> <SVO> V PRES -SG3 VFIN @+FMAINV
 - "<\$.>"

Egy transzformációs módszer: a Brill-tagger

❑ Transzformáció-alapú tanulás:

{címkézett korpusz}

→ Brill →

{transzformációs szabályok sorozata}

❑ A szabályok formája:

ÍRD ÁT a z címkét y-ra, HA az előző címke x

ÍRD ÁT a z címkét y-ra, HA az előző szó w,

ÉS a következő szó címkéje z

❑ A rendszer működése:

- 1) Inicializálás: a leggyakoribb címkék választása
- 2) Szabályválasztás: a „leghasznosabb” szabály választása
- 3) Szabály alkalmazása a tanító korpuszra, majd 2

A TnT egyértelműsítő

- ❑ Thorsten Brants (2002)
- ❑ Másodrendű, simított Markov-modell:

$$\operatorname{argmax}_{t_1 \dots t_T} \left[\prod_{i=1}^T P(t_i | t_{i-1}, t_{i-2}) P(w_i | t_i) \right] P(t_{T+1} | t_T)$$

$$P(t_3 | t_1, t_2) = \lambda_1 \hat{P}(t_3) + \lambda_2 \hat{P}(t_3 | t_2) + \lambda_3 \hat{P}(t_3 | t_1, t_2)$$

- ❑ Fontosabb tulajdonságai:
 - ismeretlen szavak kezelése
 - végződések levágása
 - nagy kezdőbetűk figyelembevétele

Magyar egyértelműsítők

❑ hunpos (2007)

- BME MOKK
- a TnT OCaml-ben történt nyílt forráskódú újraimplementálása
- MSD kódok

❑ MagyarLánc (2010)

- SZTE
- Stanford POS-tagger + morfológiai elemző
- MSD kódok

❑ PurePOS (2013)

- PPKE ITK
- a TnT alapján Javában (azóta Pythonban is)
- integrált morfológiai elemzővel (Humor)

Mi egyértelmű?

Az utca végén levő legelőre mentem legeltetni.

Az = **az[DET]=Az**

az[NM]=Az

utca = **utca[FN]+[NOM]**

végén = **vég[FN]+é[POS]+n[SUP]**

vég[FN]+é[PSe3]+n[SUP]

levő = **lesz[IGE]=lev+ő[MIF]+[NOM]**

legelőre = **legelőre[HA]**

legelő[FN]+re[SUB]

legel[IGE]+ő[MIF]+re[SUB]

mentem = **ment[IGE]+em[Te1]**

ment[MN]+em[PSe1]

megy[IGE]=men+tem[Me1]

megy[IGE]=men+t[MIB]+em[PSe1]

legeltetni. = **legeltet[IGE]+ni[INF]+.[STOP]**

legel[IGE]+tet[MUV]+ni[INF]+.[STOP]



Számítógépes morfológiai alkalmazások

A hibák és normák kezelése („proofing tools”)

- ☐ A helyesírási normák, a szabályok és a gépi helyesírás-ellenőrzés viszonya
- ☐ A helyesírási szabályzatok viszonya a gépi helyesírás-ellenőrzéshez
- ☐ Formai és jelentéstani szempontok gépi kezelése
- ☐ A gép a „nyelvhelyességi tanácsadó” szerepében
- ☐ A nehezen értelmezhető elemzések okozta problémák
- ☐ Szószinten javítható hibák: betűhibák és elütések, hibás egybeírások, hibás kötőjelezés, szóismétlések (!)
- ☐ Szószinten nem javítható hibák: hibás különírások, központoszási hibák, egyeztetési hibák, szórendhibák, vonatkozó névmások helytelen használata, stílushibák, helytelen szóhasználat, szóközök többszörözése vagy elhagyása, „értelmes” szóhibák

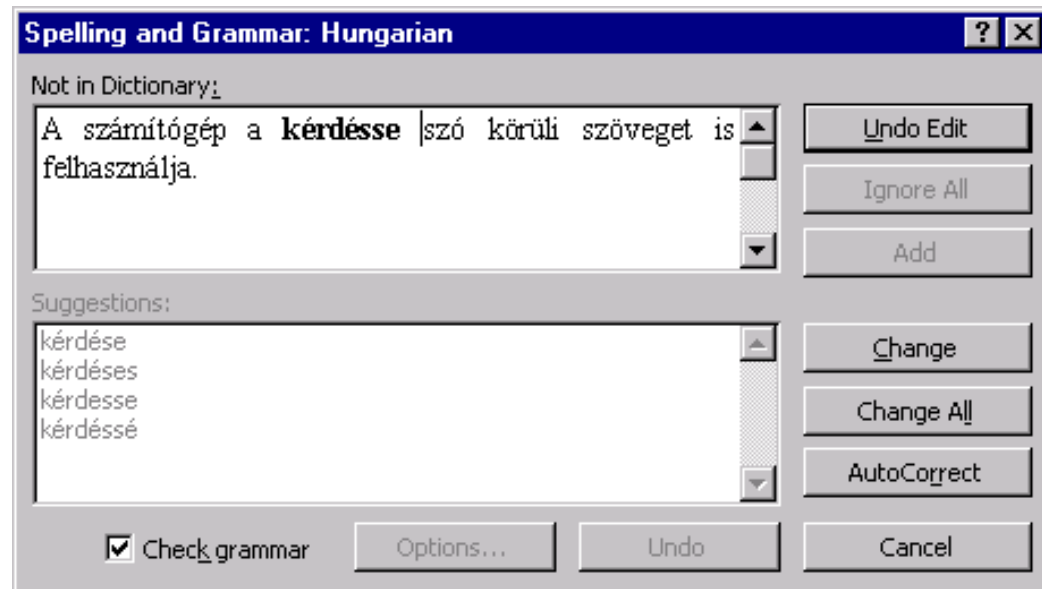
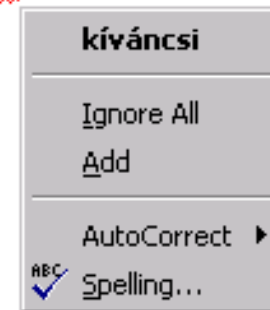
Tipikus hibák a számítógéppel létrehozott dokumentumokban

- ☐ Karakterhibák, elütések
- ☐ Valódi helyesírási hibák (általában szószintűek)
- ☐ Nyelvhelyességi hibák (általában nem szószintűek)
- ☐ Tipográfiai hibák (az átlag felhasználó nem is ismeri őket)
- ☐ A helyesírás-ellenőrzés lehetőségei és korlátai a szavak szintjén
- ☐ A szóellenőrzés és az ún. nyelvhelyesség-ellenőrzés viszonya
- ☐ A nyelvi programrendszer lehetséges hibái: a túlgenerálás (kör/kőr, alak/ház/tan + it/ít)

A szóellenőrző moduljai

- ☐ Alapszótár ill. morfológia
- ☐ Ajánlómodul és adatbázisai
- ☐ Időleges sajátyszótárak
- ☐ Kiegészítő szótár
- ☐ Kizáró szótár
- ☐ A ragozó nyelvek esetén hasznos toldalékoló sajátyszótárak lehetőségei és nehézségei

kíváncsi



Két füzér Levenshtein-távolsága

- ❑ A Levenshtein-távolság a Hamming-távolság egyfajta általánosítása
- ❑ Nemcsak egyforma hosszú füzérekre, és nemcsak betűcsere van az átalakításban, hanem beszúrás és törlés is
- ❑ Pszeudokódja:

```
int LevenshteinTavolsag(char str1[1..lenStr1], char str2[1..lenStr2])  
  // d: egy lenStr1+1 sorból és lenStr2+1 oszlopból álló mátrix  
  declare int d[0..lenStr1, 0..lenStr2]  
  // i, j: a str1 és str2 feletti iterációhoz  
  declare int i, j, cost  
  for i from 0 to lenStr1  
    d[i, 0] := i  
  for j from 0 to lenStr2  
    d[0, j] := j  
  for i from 1 to lenStr1  
    for j from 1 to lenStr2  
      if str1[i] = str2[j] then cost := 0  
      else cost := 1  
      d[i, j] := minimum(d[i-1, j] + 1,           // törlés  
                          d[i, j-1] + 1,          // beszúrás  
                          d[i-1, j-1] + cost)     // csere  
  return d[lenStr1, lenStr2]
```

Damerau-Levenshtein-távolság

- ❑ A Hamming-távolság egy másik általánosítása
- ❑ Minimális számú beszúrás, törlés, betűcsere mellett helycsere is van
- ❑ A Damerau-Levenshtein-távolság két füzér között a Levenshtein-számításból úgy kapható, hogy a fő ciklust kiegészítjük ezzel:
if($i > 2$ and
 $j > 2$ and
 $str1[i-1] = str2[j-2]$ and
 $str1[i-2] = str2[j-1]$)
 then $d[i,j] := \text{minimum}(d[i,j], d[i-2,j-2] + \text{cost})$ // *transzpozíció*
- ❑ Damerau annak idején (1964) azt is állította, hogy az emberi elütések 80%-a ezekkel korrigálható
- ❑ Ugyanakkor Damerau csak egyetlen karakteres elütésekkel számolt, szemben a Levenshtein-távolsággal

A szóellenőrzés menete

(1) Morfológiai elemzés

kérdesse ☞ <nincs ilyen szó a magyarban>

(2) Ajánlás

törlés:

érdesse, krdesse, kérésse, kédesse, kérdése, kérdéss

helycsere:

ékrdesse, krédesse, kérédsse, ..., kérdéses

nyelvspecifikus csere:

kérdéssé, kérdésse, ...

...

(3) Ellenőrzés a generátumok morfológiai elemzésével

kérdése, kérdéses, kérdésse, kérdéssé

Szóellenőrzés morfológiával

kérdése

kérdés[FN]+e[PSe3]

főnévi

kérd[IGE]+és[IF]+e[PSe3]

főnévi

kérdéses

kérdéses[MN]

melléknévi

kérdés[FN]+es[SKEP]

melléknévi

kérd[IGE]+és[IF]+es[SKEP]

melléknévi

kérdesse

kérd[IGE]+es[MUV]+se[TPe3]

igei

kérdéssé

kérdés[FN]+sé[FAC]

főnévi

kérd[IGE]+és[IF]+sé[FAC]

főnévi

Nyelvhelyesség-ellenőrzés a szóhatáron túl

- ☐ Lehetséges-e mondat szintű helyesírás-ellenőrzés?
- ☐ „Grammar checker” ?
- ☐ Parciális elemzések
- ☐ Hiba-nyelvtan vs. „szokásos” nyelvten
- ☐ Hibaelemzések, a hibák súlyozása
- ☐ A hiba és a nem-hiba határának elmosódása
- ☐ A nyelvi vagy a formai természetű hibák szűrésének preferálása
- ☐ Stílusellenőrzés számítógéppel
- ☐ Környezetfüggő szóellenőrzés (pl. **a ház mellett**)

A mondat szintjén felismerhető hibajelenségek

- ☐ Névelő-egyeztetés: *Az kutya ugat.*
- ☐ Vesszőhiány: *Nem látok vak vagyok.*
- ☐ Algoritmikus szóösszetétel: *hat lábú*
- ☐ Szemantikus szóösszetétel: *kecske béka*
- ☐ Hibás szóösszetétel: *azután a nő után*
- ☐ Hiányos szerkezetek: *Megy hasú ebédelni.*
- ☐ Téves szóhasználat: *Egyenlőre nincs mit tenni.*
- ☐ Idegen szavak szűrése: *Elromlott a printer.*
- ☐ Terjengős kifejezések: *büntetést eszközöl*
- ☐ Trágár szavak szűrése: *Le van ***va.*
- ☐ Szóközhiány, -felesleg: *Nem ,én nem akarom.*

Mondatellenőrzés a gyakorlatban

Aki másnak vermet és maga esik bele.

Elment mellettem anélkül hogy köszönt volna.

A **hog**y kötőszó elé vesszőt kell írni!
Javaslat: **mellettem,**

Ignore Sentence

 Grammar...

Ha ez egy összetett mondat, a jelzett helyen
vagy környékén egy vessző hiányzik.
Ellenőrizze!
Javaslat: **maga,**

Ignore Sentence

 Grammar...

Spelling and Grammar: Hungarian

Grammatical error:

Elszeretném adni az autómat.

Suggestions:

Az igekező nem a segédigéhez tartozik.
Javaslat: **El szeretném.**



☒ Check grammar

Options...

Undo

Helyesírási Szabályzat

File Edit Bookmark Options Help

Contents Search Back Print

131.c A magyar helyesírás szabályai 11. kiadás

Az igekező külön szó marad, ha közte és igeje (vagy igeeneve) között más szó is van: *el ne késs, le nem tenné, újjá is építjük, közre kell bocsátani, haza szabad menni, föl sem véve* stb. Hasonlóképpen: *be nem avatkozás, meg nem értés* stb.

Az igekezők különírása vagy egybeírása gyakran értelmi különbségre is utal: annak jelölője, hogy az igekező a közvetlenül utána álló igehez tartozik-e, vagy az ezt követőhöz. Más és más tehát: *megvan húszéves*, de: *meg van töltve*; *megfogom a lepkét*, de: *meg fogom nézni*; *leszokott a dohányzásról*, de: *le szokott utazni vidékre*; *beleszeretett a lányba*, de: *bele szeretett volna szólni*; *elleszek nélküle*, de: *el leszek fáradva*; *megtudták a hírt*, de: *meg tudták javítani*; stb.

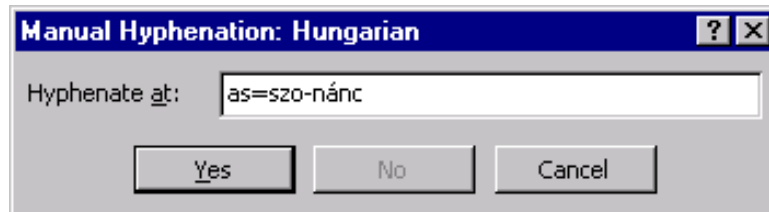
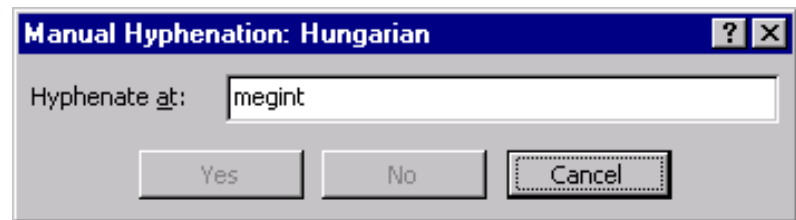
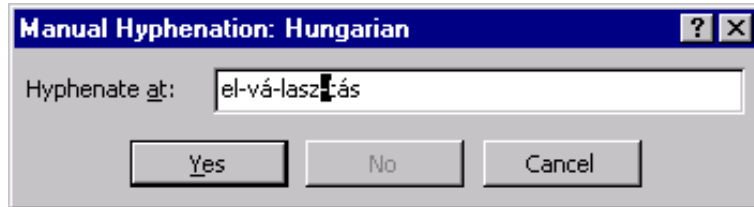
Automatikus szövegelválasztás

- ☐ Az elválasztás alkalmazása
- ☐ Automatikus és interaktív elválasztó módszerek
- ☐ Az elválasztás alapszabályai
- ☐ A morfológiai felülbírálás kérdése
- ☐ Alternatív elválasztások kezelése
(a többértelműségek, illetve a szabályok „engedékenysége” miatt)
- ☐ Szótagolás és elválasztás: a tipográfiai szokások hatása az elválasztásra
- ☐ Különleges elválasztások (hosszú kettős mássalhangzók, mássalhangzó-háromszorozódás) helyes kezelése

A magyar elválasztás alapszabályai

Alap	Elválasztva	Példa
#VV	#VV	<u>autó</u>
#VC	#VC	<u>al</u> aki
VV#	VV#	haz <u>a</u>
VV	V-V	<u>ba</u> -uxit
VC_1C_2V	VC_1-C_2V	<u>er</u> -kély
VC_iC_iV	VC_i-C_iV	vet <u>t</u> em
VCc_1c_2V	$VC-c_1c_2V$	mor <u>-zsa</u>
Vc_1c_2CV	Vc_1c_2-CV	<u>asz</u> -tal
$Vc_{11}c_{12}c_{21}c_{22}V$	$Vc_{11}c_{12}-c_{21}c_{22}V$	<u>tarisz</u> -nya
$Vc_1c_1c_2V$	$Vc_1c_2-c_1c_2V$	össze/ösz-sze

Elválasztási problémák



Más nyelven nehezebb a dolog (pl. angol):

- Oxford Advanced Learners' Dictionary (1989)

hyp-not-ize, hy-po-thesis, pro-vi-sional

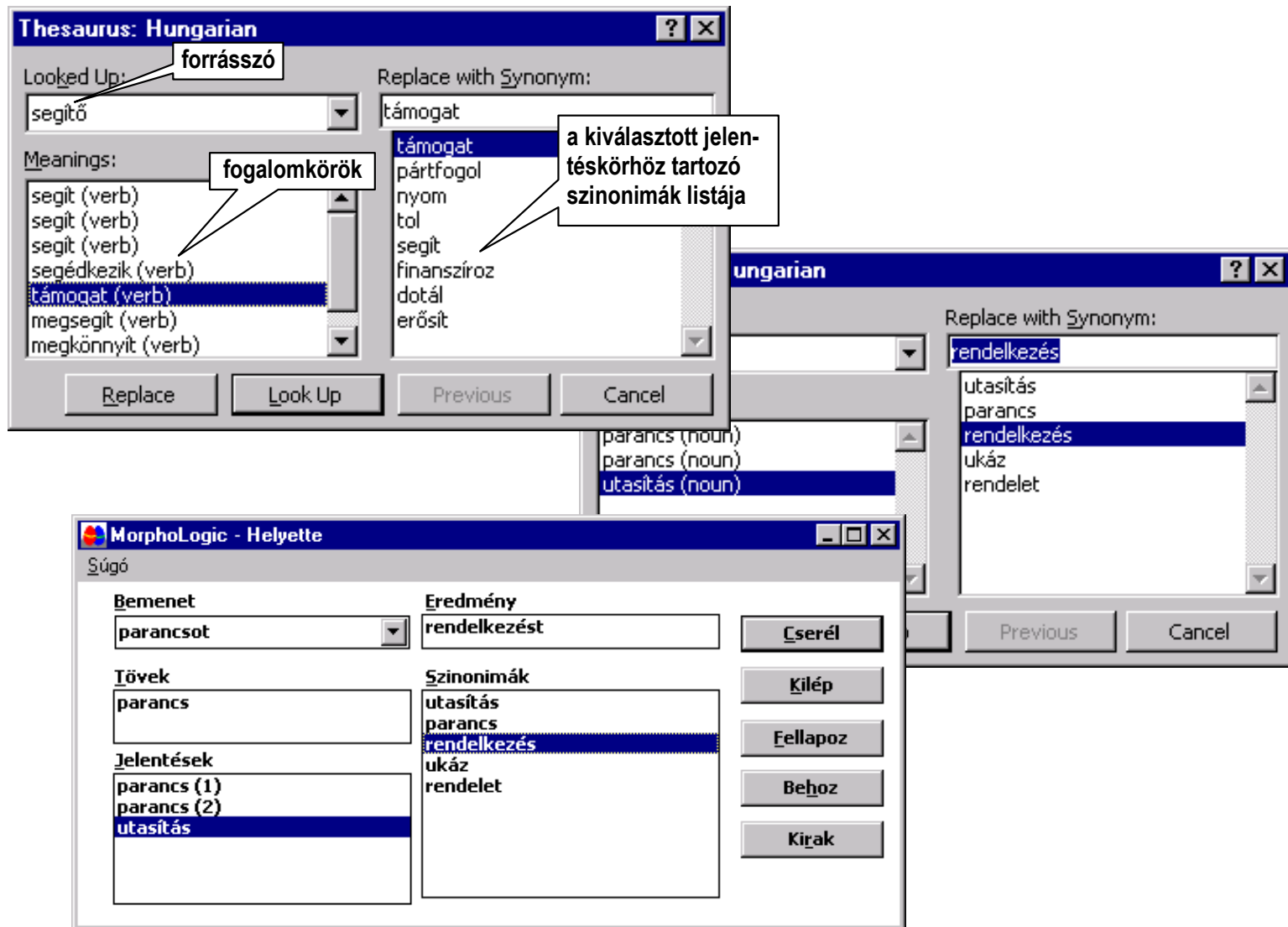
- Longman Dictionary of Contemporary English (1995)

hyp-no-tize, hy-poth-e-sis, pro-vi-sion-al

Számítógépes szinonimaszótárak és tezauruszok

- ☐ A szinonimákról
- ☐ Szinonimaszótár vagy tezaurusz?
- ☐ Tárolási és keresési problémák
- ☐ A rokon értelműség definíciója
- ☐ Az automatikus csere problémái
- ☐ Tő-visszaállítás
- ☐ Többértelműségek kezelése
- ☐ A lexikai és a szintaktikai szó különbségéből adódó nehézségek
- ☐ Az összetett szavak szinonimáinak problémája
- ☐ Morfológiai generálás minta alapján

Szinonimaszótárak



Thesaurus: Hungarian

Looked Up: **forrásszó** segítő

Replace with Synonym: támogat

Meanings: **fogalomkörök**

- segít (verb)
- segít (verb)
- segít (verb)
- segédkezik (verb)
- támogat (verb)**
- megsegít (verb)
- megkönnyít (verb)

a kiválasztott jelentéskörhöz tartozó szinonimák listája

- támogat
- pártfogol
- nyom
- tol
- segít
- finanszíroz
- dotál
- erősít

Buttons: Replace, Look Up, Previous, Cancel

ungarian

Replace with Synonym:

rendelkezés

- utasítás
- parancs
- rendelkezés**
- ukáz
- rendelet

Buttons: Previous, Cancel

MorphoLogic - Helyette

Szó

Bemenet: parancsot

Eredmény: rendelkezést

Tövek: parancs

Szinonimák:

- utasítás
- parancs
- rendelkezés**
- ukáz
- rendelet

Jelentések:

- parancs (1)
- parancs (2)
- utasítás**

Buttons: Cserél, Kilép, Fellapoz, Behoz, Kirak