Deep learning alapú szótagolás

Németh Gergely Dániel

Abstract

Kivonat

1. Bevezetés

2. Szótagoló programok

Az elterjedt szótagoló algoritmusok kétféle csoportba bonthatóak: szabály- vagy szótáralapúak. A szabad szoftverek világában *de facto* a TEXszótagolási algoritmusát használják.

A TEX eredeti szótagoló algoritmusát Prof. Knuth tervezte 1977 nyarán[1]. Ez három fő szótagolási szabályt alkalmazott: 1) utótag leválasztás, 2) előtag leválasztás és 3) magánhangzó - mássalhangzó - mássalhangzó - magánhangzó (vccv) elválasztás, azaz ha ilyen betűnégyes található a szóban, legtöbb esetben a mássalhangzók mentén elválasztható. Ez a három szabály gyakran alkalmazható, azonban már az első algoritmusban kiegészült kisebb szabályokkal („break vowel-q" vagy „break after ck") és kivételek listájával(300 szó).

A TEX82 verzióhoz megjelent Liang szótagoló algoritmusa[2], aminek legfontosabb újítása a minta(*patterns*) alapú szótagolás bevezetése volt. Ennek lényege, hogy a szótagolási szabályok mintákra definiálódtak és az algoritmus a szótagolás során ezeket a mintákat keresi a szóban.

A TEXjelenlegi verziójában a Hunspell szótagoló algoritmust alkalmazzák[3]. Ez az eljárás Liang algoritmusán alapszik, amit nyelvfüggő speciális szótagolási kiterjesztésekkel (*non-standard hyphenation extension*) egészít ki.

2.1. Liang algoritmusa

Liang szótagoló algoritmusának alapját a szótagolási minták alkotják[2]. Az algoritmust az angol `hyphenation` elválasztását az alábbi módon végzi:

Az algoritmus először megnézi, hogy a szó benne van-e a kivételek listájában (ez lényegében teljes szavakat tartalmaz mintaként). A *hyphenation* szó nincs a kivételek listájában.

Ezután a szó elejére és végére illeszt egy pont jelző karaktert. Ennek jelentősége azoknál a mintáknál lesz, amelyek akkor érvényesülnek, ha a szó elején, vagy a végén szerepelnek.

```
.hyphenation.
```

Ezt követően a minták között keres illeszkedést. A *hyphenation* szóra ezek az alábbiak: `hy3ph, he2n, hena4, hen5at, 1na, n2at, 1tio, 2io` [2, 37. oldal] A megfelelő mintákat ráillesztve a szóra, a bennük szereplő számokat a szó karakterei közé szúrva az 1. ábrán szereplőket kapjuk.[1]

```
. h y p h e n a t i o n .
  h y3p h
        h e2n
        h e n a4
        h e n5a t
           1n a
            n2a t
               1t i o
                 2i o
_____________________
.0h0y3p0h0e2n5a4t2i0o0n0.
  h y-p h e n-a t i o n
```

1. ábra. A *hyphenation* szótagolása

Az algoritmus következő lépésében minden két szomszédos karakter közé illesztünk egy számot. Alapértelmezetten 0-t és ha ennél nagyobb számot találtunk a mintaillesztésnél, a legnagyobb kapott értéket adjuk meg.

A szótagolás szabálya innen már csak egy lépés: a páratlan számok mentén elválasztunk, a párosoknál nem. Ezzel megkaptuk a `hy-phen-ation` elválasztást.

2.1.1. Minták választása

A fenti algoritmus hatékonyságát nyilvánvalóan a minták mennyisége és hatékonysága határozza meg. Csak azok a szavak választódnak el, amelyekre illeszkedik minta és csak azok lesznek jó elválasztások, melyekre jó minta illeszkedik.

2.2. Hunspell

2.2.1. Szótagolási hibák a magyar Hunspellben

Hunspell szótagolása:	Helyesen:
au-tó-val	a-u-tó-val

3. Deep learning ismertető

Hivatkozások

[1] Donald Ervin Knuth. *TEX and METAFONT: New directions in typesetting*. American Mathematical Society, 1979.

[2] Franklin Mark Liang. *Word hyphenation by computer*. Department of Computer Science, Stanford University, 1983.

[1] Az ábrázolásmód Németh cikkéből származik[3].

[3] László Németh. Automatic non-standard hyphenation in openoffice. org. *COMMUNICATIONS OF THE TEX USERS GROUP TUGBOAT EDITOR BARBARA BEETON PROCEEDINGS EDITOR KARL BERRY*, page 32, 2006.