Podela reči crticom na kraju reda u tekstu

Projekat iz Sistema baziranih na znanju

Student: Vasilije Marković SV15/2021

Motivacija

Automatski prelom teksta je od suštinskog značaja u procesiranju prirodnog jezika, izdavaštvu, mobilnim aplikacijama i alatima za obradu dokumenata. Tradicionalni algoritmi uglavnom koriste jednostavna pravila o dužini reda ili razmaku između reči, ali često zanemaruju slogovna pravila i jezičke specifičnosti. To dovodi do estetski nepravilnog teksta, teško čitljivih redova ili nepravilnog deljenja reči. Cilj ovog projekta je da se napravi inteligentni sistem za optimizovano i jezički korektno formatiranje teksta.

Pregled problema

U literaturi se problem slogovne podele u srpskom jeziku obrađuje u radu [1], gde je formiran algoritam za raščlanjivanje reči na slogove. Iako taj pristup određuje granice slogova sa visokom preciznošću, on nije dovoljan za razdvajanje reči crticom u tekstu, jer ne uključuje određena morfološka i stilska pravila. Na primer, iako je po ljudskom jeziku intuitivnija podela *pre-duslov*, konvencija pisanja teksta preferira *pred-uslov* zbog očiglednog semantičkog razdvajanja složenice [2, str. 305].

Pored toga, većina programa uopšte ne nudi ili nudi nepotpunu ugrađenu funkcionalnost prelamanja reči za srpski. Korisnici su zato često prinuđeni da koriste ruske ili bugarske šablone dok pišu na ćirilici, što može dovesti do nepravilnih preloma. Ovaj nedostatak je naročito problematičan u oblastima gde je formatiranje ključno, poput prevođenja filmskih titlova ili stripskih dijaloga u balončićima.

Prednost ovog projekta je ta što nudi rešenje koje kombinuje slogovna, morfološka i stilska pravila, sa obradom u realnom vremenu.

Metodologija rada

Ulazna tačka sistema zasniva se na Complex Event Processing mehanizmu. Tekst na kom se primenjuju pravila se ne posmatra kao gotova celina, već kao tok događaja koji pristižu jedan po jedan. Svaka korisnikova akcija dodavanja ili brisanja simbola u tekstu prouzrokuje pozivanje određenog događaja.

CEP koristi *length-based sliding window*, odnosno, pamti konačan broj najskorijih događaja. Pošto korisnik ima opciju da dodaje ili briše karaktere, CEP treba da bude podešen da pamti **L + 1** najskorijih događaja i detektuje prekoračenje reda kada se pojavi najmanje **L** unešenih karaktera (**LetterEvent** ili **LineBreakEvent**), oduzimajući zapamćeni **offset**, odnosno koliko

poslednjih događaja se vezuje za prethodni red. Kada se detektuje prekoračenje reda, to znači da je reč u kritičnoj sekciji i aktiviraju se pravila za raščlanjivanje te reči na slogove.

Nakon što se identifikuju potencijalne slogovne granice, pokreću se pravila za razdvajanje reči u tekstu crticom. Ona filtriraju da li neka od tih granica ispunjava uslove da bude iskorišćena za umetanje crtice. Veću prednost imaju granice bliže kraju reči nego početku, kako bi što veći deo reči ostao u gornjem redu. Ako postoji validan kandidat, reč se na tom mestu razdvaja između redova, a u suprotnom se cela reč prenosi u sledeći red bez umetanja crtice.

Kada stigne **SpaceEvent**, reč je zatvorena: njen slogovni raspored i mogući prelomi više se ne menjaju, a fokus prelazi na sledeću reč. Na taj način sistem kontinuirano održava ažurnu sliku o trenutnom redu i poslednjoj reči, donoseći odluke o prelomu reda u realnom vremenu, umesto da čeka unos celog teksta.

Ulazi u sistem

Događaji:

- **LetterEvent** – poziva se na svaki pritisak alfanumeričkog tastera
- **SpaceEvent** – poziva se unosom razmaka
- **LineBreakEvent** – poziva se prelaskom u novi red

Podaci:

- Trenutno poslednji red teksta
- **L** – zadata maksimalna dužina svakog reda u tekstu
- Skup pravila za razdvajanje reči na slogove
- Skup pravila za odvajanje reči
- Lista čestih prefiksa

Izlazi iz sistema

Izlaz iz sistema je pravilno formatiran tekst u kom su ubačena mesta gde su optimalno ubačene crtice za prelom reči. Korisniku je tekst prikazan u redovima tako da nijedan ne prelazi maksimalnu dužinu teksta **L** i dodate crtice, ako ih ima, nalaze se na kraju redova.

Pošto sistem obrađuje ulaz u realnom vremenu, svaka nova promena može da utiče samo na poslednji red. Svi prethodni redovi se smatraju „zatvorenim“ i njihov sadržaj ostaje nepromenjen. Zbog toga izlaz sistema mora da čuva sve prethodno formirane redove u posebnoj strukturi, dok se samo poslednji red dinamički ažurira u skladu sa novim događajima.

Baza znanja

Prva pravila definišu uslove pod kojima se pokreće čitav rezon za određivanje pozicije crtice.

Pravilo A10: Kada CEP registruje **EndEvent**, na kraj teksta se dodaje pritisnuti interpunkcijski znak.

Pravilo A11: Kada CEP registruje **LetterEvent**, na kraj teksta se dodaje pritisnuti simbol i poziva se pravilo **A22**.

Pravilo A21: Ako poslednji red u tekstu ima samo jednu reč i dužina prethodnog reda je najviše **L - 2** (odnosno postoji mogućnost da bar jedno slovo iz poslednje reči treba prebaciti u prethodni red), poslednji red se lepi na kraj pretposlednjeg i aktivira se pravilo **A22**.

Pravilo A22: Ako dužina poslednjeg reda prevazilazi zadatu dužinu **L**, aktiviraju se pravila **B11** i **B12** na svakom slovu te reči.

Zatim, pravila za razlaganje reči na slogove definisana su u vidu algoritma u radu [1].

Pravilo B11: Ako slovo pripada skupu (*a*, *e*, *i*, *o*, *u*), njegova pozicija se beleži kao jezgro sloga – **Nucleus**.

Pravilo B12: Ako je slovo *l* ili *n;* ili *r* koje nije praćeno sa *je*, beleži se kao potencijalno jezgro – **NucleusCandidate** i na njemu se aktiviraju pravila **B21**, **B22** i **B23**.

Pravilo B21: Ako je slovo **NucleusCandidate** i nalazi se između dva suglasnika manje sonornosti, njegova pozicija se beleži kao **Nucleus**. (primer: *krv*)

Pravilo B22: Ako je slovo **NucleusCandidate** i nalazi se na početku reči i praćeno je suglasnikom manje sonornosti, njegova pozicija se beleži kao **Nucleus**. (primer: *rđa*)

Pravilo B23: Ako je slovo **NucleusCandidate** i nije *r* i nalazi se na kraju reči iza suglasnika manje sonornosti, njegova pozicija se beleži kao **Nucleus**. (primer: *bicikl*)

Kada se obeleže sva jezgra u reči, za svako od njih se pokreću pravila koja određuju koji suglasnici stoje uz njega, odnosno gde su precizne granice slogova.

Pravilo B30: Ako je jezgro najdešnje u reči, posle njega se ne stavlja granica (ovo pravilo radi i za reči sa samo jednim slogom).

Pravilo B31: Ako je jezgro praćeno dvoma sonantima (*v*, *j*, *l*, *lj*, *m*, *n*, *nj*, *r*) tako da drugi nije *j* praćeno se *e*, slogovna granica se stavlja između ta dva sonanta.

Pravilo B32: Ako je jezgro praćeno dvoma sonantima (*v*, *j*, *l*, *lj*, *m*, *n*, *nj*, *r*) tako da drugi jeste *j* praćeno se *e*, slogovna granica se stavlja pre sekvence suglasnikâ

Pravilo B33: Ako je jezgro praćeno nazalom (*m*, *n*, *nj*) ili plozivom (*p*, *b*, *t*, *d*, *k*, *g*); koji je dalje praćen bilo kojim suglasnikom koji se ne nalazi u skupu (*v*, *j*, *l*, *lj*, *r*), slogovna granica se stavlja između ta dva suglasnika.

Pravilo B34: Ako za jezgro nisu zadovoljena pravila **B31**, **B32** i **B33**, slogovna granica se stavlja odmah iza jezgra.

Nakon što je reč podeljena na slogove, pravila iz *Pravopisa srpskoga jezika* [2, str. 303] nalažu pod kojim uslovima se ona sme prelomiti u tekstu. Ako nijedna od granica ne zadovoljava uslove, cela reč prelazi u sledeći red.

Rezon nalaže da se kreće se od najdešnje granice koja ne prelazi **L** i ona koja bezbedno prođe kroz naredna tri pravila, postaje mesto crtice. Time se postiže da što manji deo reči bude odvojen u sledeći red. Za ostvarivanje ovog efekta koristi se *accumulate* funkcija koja nalazi *max* vrednost pozicije mogućih granica.

Pravilo C21: Ako granica deli reč usred detektovanog prefiksa ili infiksa (infiks npr. u slučaju *potpredsednik* u kom i *pot-* (*pod-*) i *pred-* imaju značenje), reč se ne sme prelomiti na mestu granice.

Pravilo C22: Ako je bilo koji od dva dela levo i desno od slogovne granice sačinjen samo od suglasnika, reč se ne sme prelomiti na mestu granice.

Pravilo C23: Ako granica razdvaja dva samoglasnika, reč se ne sme prelomiti na mestu granice.

Pravilo C24: Ako granica ne zadovoljava nijedno od pravila **C21**, **C22** i **C23**, tekst dobija crticu na njenom mestu i sve iza nje prelazi u novi red.

Pravilo C25: Ako nije nađena pogodna granica, reč prelazi u novi red.

Korišćenje *Template*-ova

Template pravila koristiće se na mestima gde se ponavljaju isti obrasci rezonovanja, ali sa različitim vrednostima parametara radi smanjivanja dupliranog koda i povećanja fleksibilnosti sistema. U ovom projektu su pogodni za naredne funkcije:

- Pri instanciranju maksimalne dužine reda **L,** tako da se ista logika može primeniti za različite širine teksta
- Pri detekciji slogovnih jezgara, gde se isti obrazac pravila primenjuje na različite samoglasnike (a, e, i, o, u) i suglasnike-kandidate (l, n, r), kao i beleženju odnosa veća--manja sonornost između slova

- Analogno, pri preciznom određivanju slogovnih granica gde za različite grupe suglasnika tretiraju na isti način
- Pri dodavanju semantičkih granica za prefikse, gde se isti obrazac koristi za sve prefikse iz unapred definisane liste čestih prefiksa

Primer rezonovanja

Ilustrujmo rad sistema na konkretnom jednostavnom primeru. Neka korisnik unosi rečenicu *„Hleb je postan obrok.“* sa maksimalnom dužinom reda **L = 11** simbola.

Dok god korisnik ne stigne do slova *t*, CEP neće inicirati algoritam za razlaganje na slogove (*Hleb je pos* sadrži 11 simbola). Dok je poslednja reč *post*, jedino nađeno jezgro je *o* i pravilo **B30** nalaže da se posle njega ne stavljaju granice, te cela reč *post* prelazi u sledeći red i korisnik vidi:

Hleb je
post

Dodavanjem slova *a* reč postaje dvosložna i nađena jezgra su na samoglasnicima *o* i *a*. Slovo *o* je praćeno dvoma suglasnicima *s* i *t* koja zadovoljavaju pravilo **B34** i postavljena granica deli reč odmah iza jezgra, odnosno na *po*|*sta.*

Algoritam se završava pravilom **C24** i crtica je stavljena na njeno mesto. Korisniku se prikazuje sledeći razlomljeni tekst.

Hleb je po-
sta

Primetimo da crtica zauzima jedno mesto simbola i zato prvi red koji sada glasi *Hleb je po-* sadrži 11 simbola. Nakon što korisnik unese *n*, algoritam se pozove nad *postan* i proizvede identičnu granicu. Sada je na ekranu ispisana cela reč sa crticom na istom mestu kao ranije.

Hleb je po-
stan

Korisnik nastavlja sa unosom razmaka i poslednje reči *obrok*, koja nije dovoljno dugačka da prevagne donji red (zajedno sa prelomljenim *stan*, tačkom i razmakom, zauzima tačno 11 mestâ). Na kraju se vidi ceo tekst.

Hleb je po-
stan obrok.

Literatura

[1] M. Marković, A. Kovač, *A Rule-Based Syllabifier for Serbian*, Department of English Language and Literature, Faculty of Philosophy, University of Novi Sad & Department of Language Science and Technology, Saarland University.

[2] M. Pešikan, J. Jerković, M. Pižurica, *Pravopis srpskoga jezika*, Novi Sad, Serbia: Matica srpska, 1994.