Uporaba algoritma ROVER pri razpoznavanju slovenskega govora

Tomaž Rotovnik, Mirjam Sepesy Maučec, Bogomir Horvat

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko
Smetanova ul. 17, 2000 Maribor, Slovenija
tomaz.rotovnik@uni-mb.si

Povzetek

V članku obravnavamo algoritem za zmanjšanje napake razpoznavanja pri uporabi več različnih razpoznavalnikov (ROVER – Recognition Output Voting Error Reduction). Najprej predstavimo zasnovo samega algoritma ROVER, nato splošni postopek izgradnje sistema z velikim slovarjem besed za slovenski jezik. Pri tem uporabimo podbesedne jezikovne modele, ki so bolj primerni za pregibne jezike. Po dvoprehodni strategiji razpoznavalnikov (*HVite, Trace projector* in *Julius*) uporabimo ROVER. Za primerjavo izvedemo več metod glasovanja. Uporabimo tudi novo metodo, ki vključuje glasovanje s pomočjo frekvence besed in uporabo jezikovnega modela. Triprehodna strategija razpoznavanja je v primerjavi z dvoprehodno izboljšala rezultat razpoznavanja za 3% absolutno.

1. Uvod

V zadnjih letih se procesorska moč in pomnilniška zmogljivost vseh dostopnih računalniških sistemov neprenehoma povečujeta. Na področju procesiranja (razpoznavanja) govora to pomeni uporabo vedno večjih akustičnih in jezikovnih modelov, povezanih s prefinjenimi dekodirnimi algoritmi v razpoznavalnikih z velikim slovarjem besed. Možni so postali tudi alternativni pristopi, na primer kombiniranje izhodnih hipotez več raje manj učinkovitih, toda hitrejših razpoznavalnikov tekočega govora. Najboljši tak pristop je predlagala govorna skupina z inštituta NIST (National Institute of Standards and Technology) leta 1997 in se imenuje ROVER (Fiscus, 1997). ROVER je bil prvič uporabljen pri vrednotenju sistema *LVCSR Hub 5-E* leta 1997. Napaka razpoznavanja se je v tem primeru zmanjšala za 5.5 % absolutno (Fiscus, 1997). Uspešno je bil uporabljen tudi pri vrednotenju baze *Broadcast News*, kjer se je napaka zmanjšala za 2,9 % absolutno (Pallett et al., 2000).

Algoritem ROVER lahko uporabljamo nad množico hipotez ali nad n-najboljšimi (*n-best*) seznami posameznih razpoznavalnikov (Mangu et al., 1999, Stolcke et al., 1997). Slednji za minimizacijo napak na nivoju besed uporabljajo množice besednih mrež. Prav tako se v ROVER vpeljujejo dodatne funkcije izgube (*loss functions*), ki pripomorejo k še večjemu zmanjšanju napake (Goel et al., 2000).

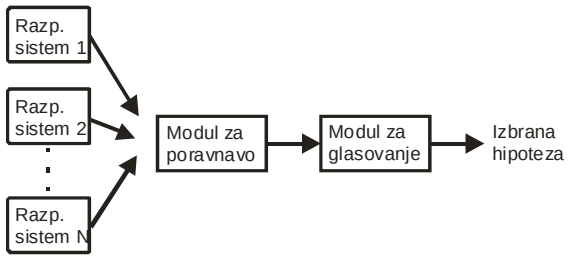
Pri uporabi algoritma ROVER je še veliko odprtih vprašanj. Eno od njih je, kako pomemben je vrstni red hipotez posameznih razpoznavalnikov pri postopku poravnave ali kolikšno je optimalno število uporabljenih razpoznavalnikov. Algoritem, opisan v tem članku, uporablja za glasovanje linearno kombinacijo frekvence besed in jezikovnega modela. Jezikovni model vpliva na uporabo pravilnega konteksta v razpoznani hipotezi.

V naslednjem poglavju je podrobno predstavljen algoritem ROVER. Zatem je na kratko opisan postopek izgradnje sistema za generiranje hipotez s posameznimi razpoznavalniki. V četrtem poglavju je predstavljena zasnova posodobljenega algoritma ROVER. Prav tako je opisan algoritem dinamičnega programiranja, ki podpira vključevanje jezikovnih modelov. Rezultati eksperimentov, kjer primerjamo različne metode glasovanja, so podani v predzadnjem poglavju.

2. Zgradba sistema

Algoritem je namenjen zmanjšanju napake razpoznavanja pri avtomatskem razpoznavanju govora. Izkorišča razlike v naravi napak posameznih razpoznavalnikov ter jih upošteva pri združitvi hipotez in njihovem nadaljnjem procesiranju. Izvaja se v dveh korakih, kot je prikazano na sliki 1:

- Najprej se hipoteze, pridobljene iz posameznih razpoznavalnikov (izhodne hipoteze), poravnajo v besedno mrežo. Pri tem se enake besede združijo, tako da se v vsakem vozlišču nahajajo različne besede.
- V modulu za glasovanje se v vsakem vozlišču izbere beseda glede na najboljši rezultat glasovanja.



Slika 1: Zgradba sistema ROVER.

Za poravnavo izhodnih hipotez je uporabljen iterativni postopek. Najprej se poravnata dve najboljši hipotezi. Za merjenje razdalj pri poravnavi se uporablja Levensteinova razdalja. K zgrajeni besedni mreži se nato poravna naslednja hipoteza in pri tem generira novo mrežo. Ta postopek se ponavlja do zadnje hipoteze. Zgrajena besedna mreža vsebuje vozlišča, v katerih se nahajajo samo različne besede. Pri postopku poravnave pride tudi do primera pojavitve dveh enakih besed v vozlišču. V vozlišču ostane samo ena beseda, ohrani pa se informacija o pogostosti pojavitve enake besede. Kadar je naslednja hipoteza daljša od prejšnjih hipotez, pride do vrivanja besed oziroma do ustvarjanja novih vozlišč. Pri tem se v novonastalo vozlišče poleg vrinjene besede doda še prazna beseda[1]. S tem se omogoči pravilno sosledje posameznih hipotez. Glasovanje v posameznem vozlišču se vrši

[1] Prazni besedi je prirejen znak @.

neodvisno od ostalih vozlišč in temelji na frekvenci pojavljanja besede v vozlišču ali na vrednosti zaupanja posamezne besede. Ker se pri glasovanju ne upošteva kontekst, lahko ima končna hipoteza zelo visoko perpleksnost to pa je v nasprotju s pristopi trenutnih razpoznavalnikov, ki uporabljajo jezikovne modele za izbiro naslednjih besed v procesu razpoznavanja. Uporaba jezikovnih modelov posredno zmanjšuje prepleksnost hipotez.

3. Generiranje hipotez

Za gradnjo akustičnih modelov smo uporabili govorno bazo SNABI, ki vključuje 52 govorcev (Kačič et al., 2000). Podbesedni jezikovnih modeli (Maučec et al., 2001) so bili grajeni s pomočjo korpusa besedil - člankov časopisa Večer od letnika 1998 do letnika 2000. Korpus obsega 60 miljonov besed. Akustični modeli so bili zgrajeni iz tristanjskih prikritih modelov Markova (HMM) z levo-desno topologijo. Za določanje parametrov smo uporabili *Baum-Welchev* algoritem. Končni akustični modeli so vsebovali 4090 stanjsko vezanih trifonskih modelov s 16 Gaussovimi porazdelitvami. Slovar izgovorjav je bil sestavljen iz 20.000 najpogostejših besed v korpusu besedil - člankov časopisa Večer.

V testni množici je bilo 7 % besed, ki jih ni bilo v slovarju. Pri generiranju izhodnih hipotez smo pri prvem prehodu uporabili bigramske jezikovne modele, pri drugem prehodu pa smo za razpoznavalnik *Hvite* uporabili trigramske (oznaka HTK-3) in štirigramske (oznaka HTK-4) jezikovne modele. Razpoznavalnik *Trace projector* je pri drugem prehodu prav tako uporabljal trigramske jezikovne modele (oznaka ISIP-3). Pri razpoznavalniku *Julius* pa smo zaradi same izvedbe iskalnega algoritma morali uporabiti obrnjene trigramske jezikovne modele (oznaka JU-R3).

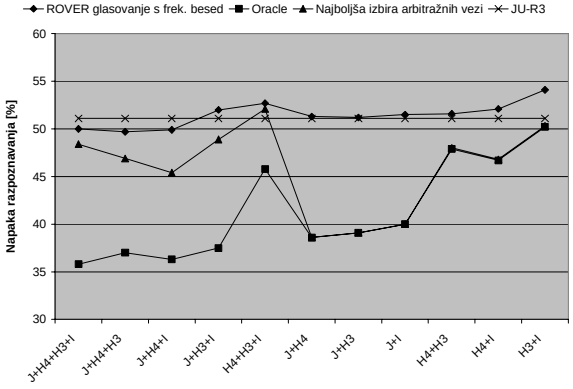
4. Eksperimenti

V tabeli 1 so podani rezultati razpoznavanja pri uporabi različnih razpoznavalnikov. Najboljši rezultat je dosegel razpoznavalnik *Julius*, zato je bila njegova izhodna hipoteza uporabljena kot prva vhodna hipoteza za sistem ROVER. K tej hipotezi je bila poravnana izhodna hipoteza razpoznavalnika z drugim najboljšim rezultatom (HTK–4). Sledili sta še hipotezi HTK-3 in ISIP-3.

Tabela 1: Napaka razpoznavanja pri uporabi različnih razpoznavalnikov.

Razpoznavalnik	Napaka [%]
Julius (JU-R3)	**51,1**
HVite (HTK-4)	**51,5**
HVite(HTK-3)	**53,6**
Trace projector (ISIP-3)	**56,3**

Takšna razporeditev vhodnih hipotez lahko še dodatno zmanjša napako, saj pomeni večjo verjetnost pravilne poravnave. Na sliki 2 je prikazana napaka razpoznavanja v odvisnosti od kombinacije in števila uporabljenih razpoznavalnikov. Metoda *oracle* predstavlja najmanjšo možno napako razpoznavanja. S kombinacijo vseh štirih hipotez je možno doseči 64.2% natančnost razpoznavanja. Seveda je ta vrednost samo hipotetična, vendar podaja možnost zmanjšanja napake z uporabo dodatnih algoritmov in virov. Uporaba frekvenčnega glasovanja nad kombinacijo vseh hipotez je sicer zmanjšala napako razpoznavanja (1.1 % absolutno), vendar je možno izboljšati razpoznavanje še za 14.2 %. Zaradi tega smo poskušali zmanjšati napako razpoznavanja z vključitvijo drugih virov informacij, z uporabo jezikovnega modela.



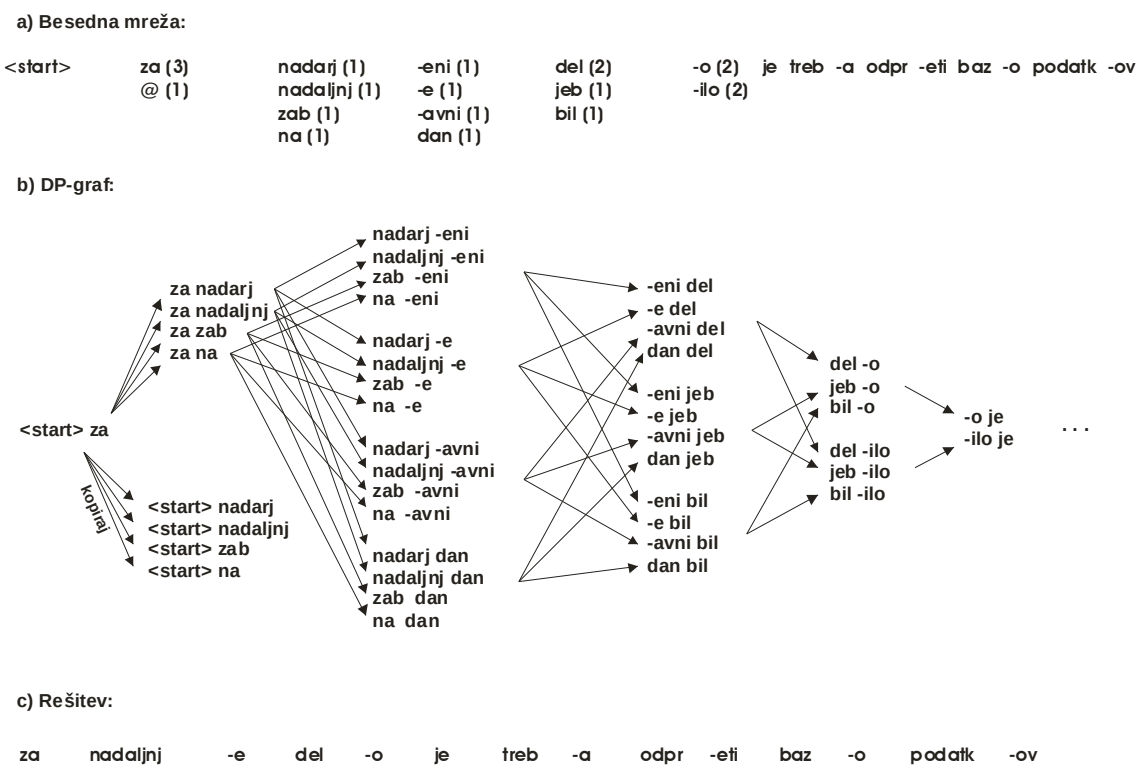
Slika 2: Odvisnost napake razpoznavanja od kombinacije in števila razpoznavalnikov.

Pri poravnavi vhodnih hipotez pogosto pride do primera, da se v besedni mreži v posameznem vozlišču pojavijo besede z enako frekvenco. V tem primeru ROVER vedno izbere besedo, ki je bila prva v vozlišču. Pri poravnavi samo dveh hipotez bo v primeru dveh različnih besed v vozlišču zmeraj izbrana beseda iz prve vhodne hipoteze. Pojavu besed z enako frekvenco pravimo arbitražna vez (*arbitrary tie*). V tabeli 2 so zbrani podatki o številu možnih arbitražnih vezi, številu arbitražnih vezi in številu navadnih vezi pri različnih kombinacijah in različnem številu razpoznavalnikov. Pod pojmom navadna vez smatramo vozlišče, ki vsebuje samo eno besedo. Napaka razpoznavanja ob upoštevanju pravilne besede v primeru arbitražne vezi je prikazana na sliki 2.

Tabela 2: Statistika vezi v besedni mreži.

Št. komb. hipotez	Št. možnih arbitražnih vezi	Število arbitražnih vezi	Število navadnih vezi
JH4H3I	**6058**	**848**	**4396**
JH4I	**5908**	**1453**	**4515**
JH3I	**5630**	**991**	**4703**
JH4H3	**5499**	**967**	**4835**
JH4	**4854**	**4854**	**5392**
JH3	**4951**	**4951**	**5227**
JI	**5171**	**5171**	**5065**
H4H3I	**3246**	**268**	**6682**
H4H3	**1900**	**1900**	**7808**
H4I	**2900**	**2900**	**6976**
H3I	**1900**	**1900**	**7764**

Arbitražne vezi je možno prekiniti z uporabo linearne kombinacije glasovalnih metod. Tako smo v eksperimentih uporabili kombinacijo frekvenčnega glasovanja in vrednosti zaupanja, v novem algoritmu pa kombinacijo frekvenčnega glasovanja in jezikovnega modela.



Slika 3: Primer razširjenega algoritma dinamičnega programiranja

4.1. Uporaba jezikovnega modela

Postopek, ki ga uporablja ROVER, zmanjša napako razpoznavanja brez upoštevanja konteksta pri glasovanju. Teoretično ima lahko dobljena izhodna hipoteza večjo perpleksnost kot katerakoli poljubna hipoteza. To je bil dodaten razlog za vključitev jezikovnega modela v proces odločanja oziroma izbire besede. Prvi korak pri novem algoritmu je enak kot pri Roverju, torej poravnava vhodnih hipotez in gradnja besedne mreže. Tudi v drugem koraku se izbira najboljša beseda v posameznem vozlišču, izbira besede pa se vrši po posebnem algoritmu, saj je za vključitev jezikovnega modela potrebno upoštevati še dodatne predhodne besede.

4.2. Algoritem dinamičnega programiranja

Slika 3 prikazuje primer besedne mreže, del grafa dinamičnega programiranja in pripadajočo izhodno hipotezo. Številke v oklepajih povedo pogostost pojavljanja posamezne besede v vozlišču. Če bi uporabili samo glasovanje s pomočjo frekvence besed, potem bi dobili napačno rešitev. V drugem in tretjem vozlišču bi bili izbrani besedi **nadarj** in **–eni.** Vidimo pa, da druga vhodna hipoteza vsebuje pravilni besedi **nadaljnj** in **–e,** ki pa zaradi arbitražnih vezi nista izbrani. Na primeru tudi opazimo, da vse možne končne rešitve nimajo enake dolžine, kadar mreža vsebuje prazne besede. Da se izniči vpliv dolžine na končni rezultat, je pri prehodu preko prazne besede k trenutnemu rezultatu dodana konstantna vrednost *c*. Vrednost se določi na neodvisni testni množici, tako da se doseže najmanjša napaka. Izkazalo se je, da sprememba vrednosti *c* nima velikega vpliva na skupno napako. Zaradi praznih besed ni možno uporabiti standardnega algoritma dinamičnega programiranja, ker njihova prisotnost onemogoča lokalno n-gramsko ocenitev. Zaradi tega smo uporabili razširjeni algoritem dinamičnega programiranja. Algoritem se izvrši v treh korakih:

- Ustvari začetno vozlišče, tako da uporabi prvo besedo iz prvega in drugega vozlišča besednega grafa (na primer [**<start>, za**]).
- Ponavlja do zadnjega vozlišča besednega grafa:
  - Postavi nova vozlišča s kombiniranjem vseh besed v trenutnem vozlišču besednega grafa (na primer **nadarj**) in vseh desnih besed besednih parov v prej postavljenih vozliščih (na primer **za**, tako nastane novo vozlišče z besednim parom [**za**, **nadarj**]).
  - V primeru, ko obstaja več vstopnih vozlišč v trenutno vozlišče, uporabi n-gramski (v našem primeru trigramski) jezikovni model. Poišče najboljši rezultat (enačba 4.1) posameznih besednih parov med trenutnimi vozlišči in obdrži samo najboljše vozlišče (na primer v vozlišče z besednim parom [**-e, del**] pripnemo prejšnje vozlišče s parom [**nadaljnj**, **-e**], povezave k ostalim parom [**nadarj**, **-e**], [**zab**, **-e**] in [**na**, **-e**] pa opustimo).
  - Če trenutno vozlišče vsebuje prazno besedo, potem kopira vsa prejšnja vozlišča v nova vozlišča in namesto uporabe n-gramskega jezikovnega modela doda konstantno vrednost *c*.
- Potuje v nasprotni smeri do pravilne hipoteze.

Takšen razširjeni algoritem dinamičnega programiranja običajno doseže zmanjšanje kompleksnosti. Ob velikem številu ponavljajočih se vozlišč s praznimi besedami je potreben dodaten procesorski čas za kopiranje in procesiranje dodatnih vozlišč. V naših eksperimentih je bil procesorski čas za magnitudo manjši kot pa v primeru upoštevanja vseh možnih hipotez v besednem grafu. Če bi v našem primeru želeli poiskati vse možne hipoteze, bi bilo potrebno oceniti 192 hipotez. Pri uporabi razširjenega algoritma dinamičnega programiranja pa je potrebno oceniti samo 80 hipotez.

$$P(N_i) = \alpha * \frac{N_{i-2} + N_{i-1} + N_i}{Num\_hyp} + (1-\alpha) * P(W_{i,j} \mid W_{i-1,j}, W_{i-2,j}) \quad (4.1)$$

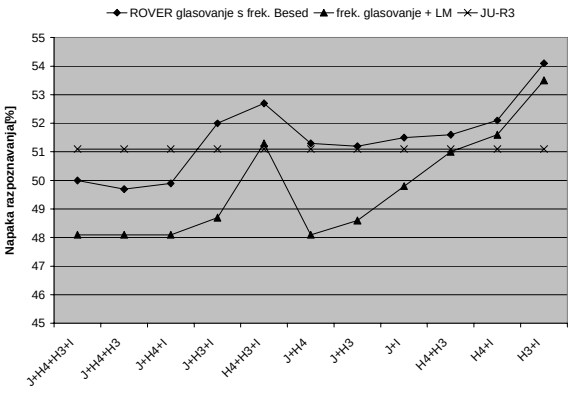
5. Rezultati

V tabeli 3 je podana napaka razpoznavanja pri sedmih metodah glasovanja in upoštevanju vseh štirih hipotez. Za referenco smo uporabili najboljšo izhodno hipotezo, dobljeno po dvoprehodni strategiji razpoznavanja (JU-R3). S to strategijo je bila dosežena napaka razpoznavanja 51,1 %. Že omenjena metoda *oracle* omogoča najnižjo napako razpoznavanja, ki jo je možno teoretično doseči z Roverjem. Ta metoda ni uporabna v praksi, ker je rezultat dosežen na podlagi pravilnih izhodnih hipotez. Iz besedne mreže se izbere tista hipoteza, ki se najbolje prilega pravilni hipotezi. Glasovanje s pomočjo frekvence besed je zmanjšalo napako razpoznavanja za 2,1 % relativno. Izvedli smo tudi eksperiment, v katerem smo za vrednosti zaupanja uporabili jezikovni model. Po pričakovanju smo dobili slabši rezultat od referenčnega zaradi napačnega ohranjanja konteksta. Ne glede na to pa je linearna kombinacija frekvenčnega glasovanja in vrednosti zaupanja zmanjšala napako razpoznavanja.

Tabela 3: Napaka razpoznavanja pri različnih metodah glasovanja.

Metoda glasovanja	α	c	Napaka [%]
Referenca (JU-R3)	/	/	**51,1**
Oracle	/	/	**35,8**
Frekvenca besed	**1**	**0,1**	**50,0**
Vrednosti zaupanja	**0**	**0,05**	**51,6**
Frek. besed + Vred. zaup.	**0,15**	**0,05**	**49,2**
Prehod z JM	/	**0,01**	**48,7**
JM samo v primeru arbitražnih vezi	/	**0,01**	**50,5**
Frek. besed + JM	**0,1**	**0,1**	**48,1**

Z naslednjo metodo smo za najboljšo hipotezo iz besedne mreže uporabili tisto, ki je imela najmanjšo perpleksnost. S tem smo dosegli drugi najboljši rezultat (48,7 %). Pri naslednji metodi smo uporabili jezikovni model samo v primeru pojavitve arbitražne vezi. Tokrat se napaka ni veliko zmanjšala. Zadnja metoda, z njo je bil dosežen najboljši rezultat, je upoštevala informacijo iz jezikovnega modela tudi pri navadnih vezeh. Enačba (4.1) prikazuje oceno besednih parov v posameznih vozliščih in se lahko posploši na poljubno velikost jezikovnih modelov. Od tega sta odvisni kompleksnost algoritma in smiselnost uporabe. Slika 4 prikazuje primerjavo standardnega algoritma Rover s frekvenčnim glasovanjem z izboljšanim novim algoritmom. Hkrati je podana tudi referenčna vrednost, dobljena z razpoznavalnikom *Julius*.



Slika 4: Napaka razpoznavanja pri različnih metodah in različnih kombinacijah razpoznavalnikov.

6. Zaključek

V članku smo predstavili zgradbo sistema ROVER. Prikazali smo problem uporabe samo frekvenčnega glasovanja in podrobneje predstavili razširjeni algoritem dinamičnega programiranja. S predlaganim novim algoritmom glasovanja smo izključili vpliv arbitražnih vezi na napako razpoznavanja. Dodatna uporaba konteksta, pridobljenega iz jezikovnega modela, je zmanjšala napako za 3 % absolutno.

7. Literatura

J. G. Fiscus. 1997. A post-processing system to yield reduced error word rates: Recognizer output voting error reduction (ROVER). V: *IEEE Workshop on Automatic Speech Recognition and Understanding*, str. 347-354, Santa Barbara, ZDA.

L. Mangu, E. Brill, in A Stolcke. 1999. Finding consensus among words: Lattice-based word error minimization. V: *Eurospeech*, str. 495-498, Budimpešta, Madžarska.

A. Stolcke, Y. König, in M Weintraub. 1997. Explicit word error minimization in n-best list rescoring. V: *Eurospeech*, str. 163-165, Rodos, Grčija.

D. S. Pallett, J. G. Fiscus, in J. S. Garofolo. 1999. 1998 broadcast news benchmark test results. V: *DARPA Broadcast News Workshop*, Washington, ZDA.

V. Goel, S. Kumar, in W. Byrne. 2000. Segmental Minimum Bayes-Risk ASR Voting Strategies. V: *ICSLP 2000*, Peking, Kitajska.

Z. Kačič, B. Horvat, in A. Zögling. 2000. Isues in design and collection of large telephone speech corpus for Slovenian language, *LREC 2000*, str. 943-946, Atene, Grčija.

M. Maučec, in Z. Kačič. 2001. Topic Detection for Language Model Adaptation of Highly-Inflected Languages by Using Fuzzy Comparison Function, V: *Eurospeech*, str. 243-246, Aalborg, Danska.