

Nenad Končar

*Translation Experts Ltd. i  
Imperial College of Science and Medicine  
London, Ujedinjeno Kraljevstvo*

Danko Šipka

*Translation Experts Ltd., London, i  
Uniwersytet im. Adama Mickiewicza  
Poznań, Poljska*

## ALGORITAMSKA I HEURISTIČKA OBRADA LEKSIČKIH PODATAKA U RAČUNALNOM PREVODIOCU NeuroTran<sup>1</sup>

Obrada leksičkih podataka (u najširem mogućem značenju tog termina) u računalnom prevodiocu NeuroTran vrši se na osnovi pravila pod nazivom gramatika minimalne informacije (skraćeno MIG, po engleskoj skraćenici za Minimal Information Grammar). Tri ideje u osnovi su programa NeuroTran i MIG-a. Prva postulira da treba minimalizirati informaciju potrebnu programu da radi, a da nove informacije program treba usvajati čitajući tekstove i komunicirajući s korisnicima, koristeći pri tom neuronske mreže. Druga je ideja na koju se oslanja NeuroTran to da treba minimalizirati napor leksikografa i tražiti od njega minimalnu informaciju, koju on već ima kao izvorni govornik. Konačno, treća ideja NeuroTrana je to da informacija kojom program raspolaže treba biti višekratno iskoristljiva i tako funkcije koje on obavlja raznorodne. Takva koncepcija iziskuje međusobno prožimanje postupaka algoritamskog tipa s onima koji su zasnovani na heurističkom principu. U radu se na primjerima konkretnih rješenja u prevođenju i usvajanju novih podataka pokazuje na koji su način povezivana ta dva postupka.

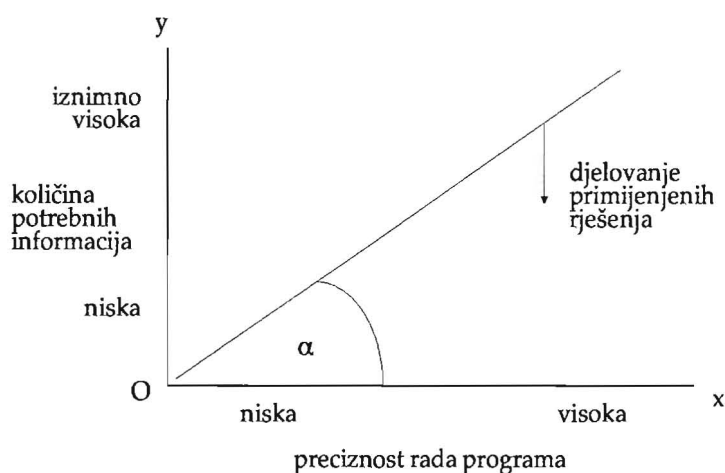
### 1. Uvod

Odnos između količine informacija koju programu treba osigurati i preciznosti rada programa za strojno prevođenje može se predstaviti dijagonalno

---

<sup>1</sup> Autori zahvaljuju poduzeću Translation Experts Ltd. za novčanu podršku, te Vladimiru Šipki i Slaveku Pavlovskom za programiranje iznesenih rješenja. Danko Šipka zahvaljuje zakladi Alexander von Humboldt za stipendiju 1990–1992. koja je omogućila razvoj gramatike upotrebnih etiketa.

monotono rastućom funkcijom. Osnovni problem bilo kojeg programa te vrste jest to što dostupni resursi, ljudski i novčani, čak i najvećih poduzeća, nikad ne mogu osigurati takvu količinu informacija koja bi omogućila visoku preciznost u radu programa. Jedna od ideja u temelju programa za strojno prevođenje NeuroTran i pravila pod nazivom MIG (od engleske kratice za Minimal Information Grammar), koja taj program rabi, jest pokušaj smanjenja kuta spomenute funkcije u odnosu na os  $x$  u smislu smanjenja količine potrebnih informacija uz istovremeno povećanje preciznosti rada. Spomenuti odnosi mogu se predstaviti kao na crtežu:



Nacrt rješenja koja bi promijenila postojeći odnos količine informacija leži u komplementarnoj primjeni algoritama, koji osiguravaju visoku preciznost u radu, ali zahtijevaju veliku količinu informacija i heuristika, koje vode ka nižoj preciznosti u radu, ali zahtijevaju daleko manje informacija.

Predstaviti ćemo najprije rješenja ugrađenja u NeuroTran i MIG uglavnom na primjeru onoga što se tradicionalno naziva gramatičkim pravilima, a nakon toga pokazati kako se ista ta rješenja primjenjuju u obradi leksičkih podataka.

## 2. NeuroTran i MIG — temeljne ideje i osnovne funkcije

Parodirajući Austina, možemo reći da NeuroTran »does things with words«. Taj program nastao u međunarodnom kompjutorskom poduzeću sa sjedištem u Londonu, Translation Experts, tipičan je postfordist, koji je istovremeno dvojezični dvosmjerni i dvomedijski (pisani i govoreni) rječnik,

rječnik sinonima, prevodilac, analizator morfološke i sintaktičke strukture, te kvalitativni i kvantitativni analizator teksta.

NeuroTran vrši sljedeće funkcije:

- pronalaženje ekvivalenata u L2 te njihovih gramatičkih i upotrebnih karakteristika,
- reprodukcija zvuka riječi,
- generiranje fleksijskih oblika riječi,
- pregledanje i izbacivanje u svežanj (datoteku) riječi koje dijele izabranu gramatičku crtu,
- pregledanje i izbacivanje u svežanj riječi s istim upotrebnim karakteristikama i/ili tematskom grupom,
- pronalaženje sinonima i antonima,
- prevođenje cijelih rečenica,
- određivanje tipa teksta i pronalaženje odgovarajućeg sinonima za taj tekst,
- analiza morfološke i sintaktičke strukture,
- kvalitativna analiza (analiza sadržaja).

Trenutno su u NeuroTranu dostupni ili se nalaze u završnoj fazi obrade sljedeći jezici: engleski (kao L1 i L2), a bosanski, češki, francuski, hrvatski, njemački, poljski, ruski, te srpski kao L1 i L2.

NeuroTran je napisan u programskom jeziku C++ za Windows, a osim na platformi Windows (3.1 i 95) dostupan je i u okruženju DOS-a, Macintosh-a te Unixa.

Tri su ideje u osnovi programa NeuroTran. Prva postulira da treba minimalizirati informaciju potrebnu programu da radi, a da nove informacije program treba usvajati čitajući tekstove i komunicirajući s korisnicima, rabeći pritom neuronske mreže. Druga je ideja na kojoj se opire NeuroTran to da treba minimalizirati napor leksikografa i tražiti od nje minimalnu informaciju, koju ona već ima kao izvorni govornik. Konačno, treća ideja NeuroTrana jest to da informacija kojom program raspolaže treba biti višekratno iskoristljiva i tako funkcije koje on obavlja raznorodne.

Pravila MIG-a podređena su tim osnovnim konceptijskim naznakama programa. Tu je i prva bitna razlika u odnosu na ustaljenu praksu uzimanja gotovog lingvističkog modela formalizacije. Ovdje je odnos jezikoslovnog modela i programa upravo obrnut. Model je razvijen isključivo za program i tom se programu podređuje.

Treba naglasiti i to da minimalizam tu ne znači minimalnu dužinu opisa, ili pak minimalnu informaciju kao takvu, nego minimalnu da bi program mogao obavljati svoje funkcije.

Prije nego što prijedemo na opis samih pravila, treba skrenuti pažnju na funkcionalnu i fleksibilnu podjelu sfere kompetencija lingvista i programera.

Pravila su, naime, čisti tekst, koji se, nakon parsiranja pretvara u oblik direktno čitljiv u programu. Na taj način lingvist može mijenjati, doradivati i dopunjavati postojeća pravila i uvoditi nova bez pomoći programera i bez ikakvih promjena u programu.

MIG minimalizira informaciju potrebnu programu tako što teži optimalnoj distribuciji informacije u pravilima s jedne i etiketama leksičkih jedinica s druge strane, te tako što koristi pravila različite klase, koja se međusobno nadopunjuju.

Ilustrirajmo to jednim jednostavnim primjerom. Da bismo u MIG-u generalizirali fleksiju hrvatske riječi *selo*, potrebne su nam dvije stvari. Prvo je *etiketa* natuknice 'selo', koja izgleda ovako:

*engleska leksička jedinica/selo, a n;/ostali hrvatski ekvivalenti*

Te pravila ovog oblika:	s ovim značenjem
CRO PARA *o,a n =>	glava pravila, koja govori da je to hrvatsko pravilo (CRO) za razvijanje paradigme (PARA) i ako se nađe natuknica s etiketom bilo što (*) o,a n, onda se (=>) u tijelu prvo govori da je to
NOUN;	imenica
NEUTER;	srednjeg roda,
O1=(1->','-1);	te da je osnova sve u etiketi do prvog zareza osim posljednjeg znaka (-1), (dakle u ovom slučaju selo-1 = sel) a da su
SINGULAR;	padeži jednine generirani tako što se
NOM=O1+o;	u nominativu na osnovu dodaje o
GEN=O1+a;	u genitivu a...
DAT=O1+u;	
ACC=O1+o;	
VOC=O1+o;	
INS=O1+om;	
LOC=O1+u;	
PLURAL;	...a padeži množine tako što se
NOM=O1+a;	u nominativu dodaje a...
GEN=O1+a;	
DAT=O1+ima;	
ACC=O1+a;	
VOC=O1+a;	
INS=O1+ima;	
LOC=O1+ima	

Pravilo iz toga primjera pokazuje prvu klasu pravila koje koristi MIG, pravila tipa konstruktor, koji spajanjem postojećeg materijala spajaju nove oblike.

Puna lista pravila i njihovih funkcija unutar MIG-a izgleda ovako:

<i>Klasa pravila</i>	<i>Funkcija</i>
<b>konstruktori</b>	koriste rječničke etikete da konstruiraju oblike koje leksem može imati
<b>mutatori</b>	mijenjaju već generirane oblike
<b>pronalazači</b>	pronalaze potrebni leksem
<b>određivači</b>	govore što je što
<b>koordinatori</b>	usklađuju jedan oblik s drugim
<b>djelitelji</b>	razbijaju veće jedinice na manje
<b>veznici</b>	vezuju manje jedinice u veće
<b>zamjenjivači</b>	zamjenjuju jedan leksem ili oblik drugim, npr. u prevodenju
<b>brojači</b>	obavljaju svu statistiku
<b>sumnjači</b>	otkrivaju situaciju gdje postoji više mogućnosti nastavka procesa
<b>kockari</b>	biraju najvjerojatnije rješenje mada i dalje postoje druga
<b>učitelji</b>	mijenjaju postojeću informaciju na osnovi podataka prikupljenih u prevodenju
<b>pričaoci</b>	komuniciraju s korisnikom programa
<b>usmjerivači</b>	određuju redoslijed izvršavanja pravila

Treba već sada posebno istaći važnost optimalne raspodjele informacije između pravila i etikete. Dobar dio alternacija, koje se u standardnom gramatičkom opisu moraju posebno obrađivati, ovdje jednostavno ne predstavlja nikakav problem, niti postoji potreba da se na njih obraća posebna pažnja. Takve su na primjer alternacija a:Ø (tzv. nepostojano a) kod imenica ili alternacije između infinitivne i prezentske osnove kod glagola, o čemu će biti riječi u dijelu o formalizaciji fleksije odgovarajuće vrste riječi. Pod alternacijama ovdje i u daljem toku teksta podrazumijevamo alterniranje osnova, dakle slučaj kad u više oblika riječ ima različite osnove, bez obzira na to da li je njihova različitost uvjetovana fonološkim, morfofonološkim alternacijama ili supletivizmom.

I za ostale alternacije opis slijedi zahtjev minimalizma. Princip je to da se piše pravilo klase konstruktor istovremeno za paradigme s alternacijama i bez njih, a da se istovremeno s njim primjenjuje mutator, koji djeluje ukoliko postoji alternacija, a ukoliko je nema, ostaje neaktiviran. Evo i primjera.

Dva pravila tipa mutator:

<i>Pravilo</i>	<i>Značenje</i>
CRO FUN CZS =>	glava, koja govori da je to hrvatska funkcija zvana CZS
CZS[O]=LAST[O][[(k,g,h)=>(c,z,s)]]	tijelo, koje govori da primjenom funkcije CZS na osnovu riječi vršimo njenu mutaciju tako što se suglasnik k,g,h na njenom kraju mijenja u c, z, s
CRO FUN CZS2 =>	isto kao u prethodnom slučaju, samo se funkcija zove CZS2
CZS2[O]=LAST[O][[(k,g,h)=>(č,ž,š)]]	i mijenja ne u c, z, s, nego u č, ž, š

i sljedeće konstruktorsko pravilo (dio je tijela ispušten, što je označeno trima točkama):

<i>Pravilo</i>	<i>Značenje</i>
CRO PARA *K,a # m => NOUN; MASCULINE; O1=(1->'); SINGULAR; NOM==O1; ... VOC=CZS2(O1)+e; ... PLURAL; NOM=CZS(O1)+i; ... DAT=CZS(O1)+ima; ... VOC=CZS(O1)+i; INS=CZS(O1)+ima; LOC=CZS(O1)+ima	glava – ukoliko se naide na odrednicu s tom etiketom, onda je to imenica muškog roda osnova je sve do prvog zareza u odrednici fleksija jednine generira se tako što se uzme osnova bez ikakvih nastavaka, a u vokativu se dodaje nastavak -e i primjenjuje funkcija CZS2 u množini se u nominativu dodaje nastavak -i te primjenjuje funkcija CZS u dativu nastavak -ima i ista ta funkcija itd.

generiraju paradigmu leksičkih jedinica kao *vojak,a # m; pjesnik,a # m;* itd. Veoma je bitno to da će funkcija izvršiti mutaciju krajnjeg suglasnika osnove ukoliko je on na listi tri suglasnika na koje pravilo tipa mutator djeluje, ali da će, ukoliko se radi o nekom drugom suglasniku konstruktor biti primijenjen, a mutator ne. Dakle, gore navedeno pravilo generirat će i odrednice kakve su: *alas,a # m; konzul,a # m;* i sl. Ta kombinacija pravila klase konstruktor i pravila

klase mutator daje bitne uštede u dužini opisa, te, što je prvobitni cilj pravila, optimizira rad programa.

Kako je već rečeno, dio alternacija rješava se već u samim etiketama i sukladno tomu u glavi pravila. Tako, npr. poljski glagol *jechać* ima sljedeću formu rječničke odrednice:

*jechać, jadę, jedziesz, iv;*

a glava i početak tijela za generiranje njegove paradigme izgledaju ovako:

POL PARA \*ć,\*ę,\*esz,iv =>

VERB;

O1=(1→>');)

INFINITIVE=O1;

ACTIVE;

PRESENT;

AFFIRMATIVE;

O1=(1→SAMEAS(1','+1))+(1','->2',');

O2=(1→SAMEAS(2','+1))+(2','->3','-2);

O3=(1→SAMEAS(1','+1))+(1','->2','-1);

SINGULAR;

FIRST=O1;

SECOND=O2+sz;

THIRD=O2;

...

Nema, dakle, potrebe za posebnim izdvajanjem izuzetka ili nekom dodatnom obradom – već sama etiketa rješava problem nepodudarnosti osnove infinitiva i prezenta.

Važna je i forma etiketa, koja pokazuje drugi aspekt minimalnosti – minimalizaciju napora od strane leksikografa. Forma etiketa takva je da ih leksikograf može staviti na osnovu svoje jezične kompetencije, bez pamćenja bilo kakvih dodatnih informacija i bez znanja lingvističkih modela. Nadalje, praksa rada na etiketama takva je da leksikograf etiketira 10% odrednica, ostatak se etiketira na osnovi koreliranja informacije dobivene u tih 10% etiketiranih odrednica sa slijedom karaktera leksičke jedinice gledanim odstrag. Radi se, naravno, o verziji već odavno poznate procedure za određivanje vrste riječi na osnovi niza znakova idući od kraja leksema, ovdje modificiranoga dvostruko: tako što se koristi informacija iz dijela etiketiranog teksta i tako što ovdje etiketa ne sadrži samo informaciju o vrsti riječi, nego i sve druge informacije potrebne za generiranje paradigme.

Kako smo na predočenim primjerima pokazali opći nacrt pravila, usredotočit ćemo se na to kako je taj skup ideja i procedura primjenjivan u obradi leksičkih podataka.

### 3. Obrada leksičkih podataka u NeuroTranu

Temeljna zadaća u obradi leksičkih podataka u tom tipu programa sastoji se u izboru odgovarajućeg ekvivalenta iz L2 za leksičku jedinicu iz L1. Kako anizomorfizam leksičkih sustava u velikom broju slučajeva stvara situacije tipa

$$\text{leksem}_1(\text{L1}) = \text{leksem}_1(\text{L2}) / \text{leksem}_2(\text{L2}) / \dots / \text{leksem}_n(\text{L2})$$

program u svakom konkretnom slučaju prevodenja s L1 na L2 mora izabrati samo jedan od mogućih oblika. (Moguće odnose i posljedice za leksikografiju posebno obrađuje Zgusta 1971.)

Koji će od ekvivalenata biti upotrijebljen, zavisit će od toga s realizacijom kojeg značenja iz polisemijske strukture leksema iz L1 imamo posla, a ono što nam može pokazati o kojoj se realizaciji radi, jest (u terminologiji Lipke 1992) *kotekst* (neposredno realizacijsko okruženje) i *kontekst* (jedinica teksta u kojoj se leksem realizira). Sukladno tome NeuroTran rabi dvije tehnike, od kojih se jedna obraća kotekstu, a druga kontekstu. Kotekstni pokazatelji u NeuroTranu obrađuju se osebujnom inačicom analize čestote, dok se kontekstni pokazatelji izvlače pomoću tzv. gramatike upotrebnih etiketa.

Gramatika upotrebnih etiketa podrazumijeva s jedne strane postojanje etikete kontekstno raslojenog leksema koja govori o tome prema kakvim kontekstima taj leksem pokazuje najveću sklonost, a s druge strane određenje o kojem se tipu konteksta radi. Prirodom stvari, etikete moraju biti prethodno osigurane, kao i skup mogućih određenja konteksta, te pravila kompatibilnosti leksemskih etiketa i kontekstnih određenja. Samo određivanje tipa konteksta ostavlja se korisniku, a ako on to ne može ili ne želi uraditi, program obračunava čestotom leksema s odgovarajućim etiketama i tako sam pokušava odrediti tip konteksta. (Osnovni pojmovi o mreži upotrebnih etiketa predstavljani su u Šipka 1994.)

U trenutačnoj fazi rada NeuroTran rabi hijerarhiziran skup etiketa identičan za lekseme i kontekste, te veoma jednostavna pravila da je kontekstna etiketa kompatibilna s identičnom leksemskom, te s ostalim leksemskim etiketama iz iste hijerarhijske grane. Evo i konkretnih primjera. Mreža upotrebnih etiketa uzima u obzir vremensko, teritorijalno i funkcionalno raslojavanje. Kako se na osnovi poznatih sociolingvističkih činjenica moglo očekivati, najraslojeniji sustav etiketa dobivamo u funkcionalnom raslojavanju. (Šire podatke o raslojavanju daje Fasold 1991.) Tako, na primjer, tekst koji je posvećen nogometnoj utakmici pripada sljedećoj hijerarhijskoj grani:

- 4 igre i zabave
- 44 sport
- 441 sportovi s loptom
- 4411 nogomet



dok tekst posvećen sudskom procesu pripada sljedećoj hijerarhijskoj grani:

- 1 znanost
- 14 društvene znanosti
- 141 pravo

Ukoliko leksičku ekvivalenciju između hrvatskog i engleskog prikazemo u pojednostavnjenom obliku sa svega dva ekvivalenta kao:

*sudac, uca m; 1...44+141/judge n; 1...141/referee n; 1...44*

*1...4411+141 – prva etiketa do znaka +: standardnojezični tekst (1), nema vremenskog raslojavanja (prva točka), nema teritorijalnog raslojavanja (druga), igre i zabave (4 na prvom mjestu iza treće točke), sport (4 na drugom mjestu); druga etiketa od znaka +: do plusa je sve kao u prvoj etiketi, a tekst je iz sfere neke znanosti (1), iz društvene znanosti (4), te prava (1)*

onda će u tekstu o nogometu kao ekvivalent biti odabrano englesko *referee*, a u tekstu o sudskom procesu *judge* zato što etikete 1...4, 1...44, 1...441, 1...4411 pripadaju istoj hijerarhijskoj grani, pa su tako i kompatibilne, dok etikete 1...1, 1...14, te 1...141 pripadaju drugoj hijerarhijskoj grani, pa su kompatibilne međusobno, ali ne i s etiketama iz prethodne grupe.

Iz toga se vidi da su algoritamskog karaktera leksemske etikete, te vrlo jednostavna pravila kompatibilnosti, dok je određivanje tipa teksta heuristička koja se oslanja ili na intuiciju korisnika ili na proceduru izračunavanja čestoće etiketa. Mada leksemske etikete program uzima kao "tvrde podatke", i one su u dobroj mjeri heuristike leksikografa, doduše potvrđivane ili odbacivane u analizama odgovarajućih kontekstnih tipova, koji prethode konačnom utvrđivanju etikete koju korisnik dobiva s programom. S druge strane, ukoliko se u kasnijem radu programa određena etiketa pokaže neučinkovitom, postoji mogućnost njene zamjene, što je jedna od funkcija klase pravila »učitelji«.

Uz već opisanu mješavinu algoritama i heuristika, količina informacija minimalizira se i tako što prikupljene informacije vrše različite funkcije. Tako mreža upotrebnih etiketa ne služi samo tome da se izabere odgovarajući ekvivalent, nego i daje mogućnost da se izbacii popis leksema vezanih uz isti kontekst (što je od velike koristi u nastavi stranih jezika), da se u kvalitativnoj analizi teksta dobije informacija o statusu tih leksema i sl. Tako se za istu informaciju dobiva više korisnih primjena, što također doprinosi smanjivanju kuta  $\alpha$  spomenutog na početku ovog rada.

Naravno, kontekstni pokazatelji ne rješavaju sve postojeće situacije donošenja odluke, pa se moraju potražiti kontekstni pokazatelji. Kontekstni se pokazatelji aktiviraju osebnim zahvatom koji podrazumijeva definiranje više-leksemskih natuknica na osovi rezultata analize čestoće. Osnovna je ideja u tome da, ukoliko leksem ima jaku vezanost uz određeni kontekst, onda će se određena kombinacija leksemskih elemenata (bez obzira na to o kakvom se sin-

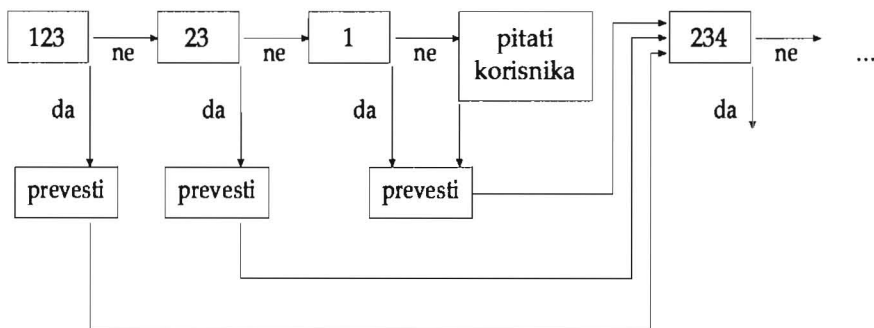
taktičkom okruženju radi) pojavljivati češće nego kad takve vezanosti nema.

Takvim kombinacijama koje pokazuju međusobnu vezanost koja premašuje određenu ustaljenu granicu, bit će dan status natuknice, bez obzira na to da li je jedinica iz leksikografske i/ili leksikološke perspektive višerječna leksička jedinica ili ne. U samom radu programa algoritam je takav da se prvo provjerava čine li tri uzastopne jedinice teksta natuknicu, pa se nakon toga provjerava to isto za dvije, pa tek onda za jednu jedinicu. Procedura izgleda kako je predstavljeno na crtežu 3.

Ako niz jedinica teksta predstavimo kao:

1 2 3 4 ... n

onda procedura odgovara na pitanje: »Postoji li ekvivalent za«



Na taj način, dobar dio situacija gdje jednorječna natuknica ima više mogućih ekvivalenata biva zaobiđena, tako što se ekvivalencija uspostavlja između trorječnih ili dvorječnih natuknica. Tako, na primjer, postoji višestruka ekvivalencija u natuknici:

*aktivan, vna, vno--;/actable--aj;/active--aj;/activated--aj;/busy--aj;/live--aj;*

ali, ako u tekstu nađemo na kombinaciju *aktivni ugljen*, taj problem je eliminiran postojanjem višerječne natuknice:

*aktivni, a, o--; ugljen, a m;/activated charcoal n;*

pa kombinacija biva prevedena prije nego što bi se mogla pojaviti situacija višestruke ekvivalencije.

Naravno, na taj način nije moguće riješiti sve vrste kontekstne vezanosti, pa je sljedeći korak izabiranje ekvivalenta koji je u cjelokupnom tekstovnom korpusu češći. Taj je zahvat zasnovan na veoma prostoj pretpostavci da se izabiranjem onog što je češće povećava vjerojatnost pravilnog izbora.

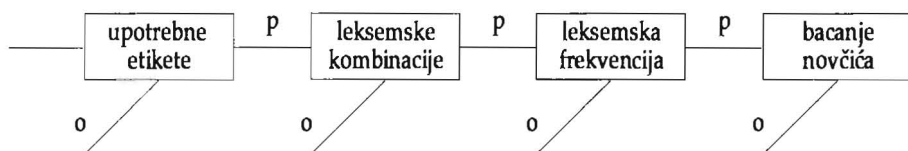
Tako, na primjer, ako imamo natuknicu (predočenu u pojednostavnjenom obliku, nakon što su upotrebne etikete eliminirale *spis, rašpa* itd. kao moguće ekvivalente):

file  $n; 1 \dots 191$ /datoteka,  $e f; N1 1 \dots 191$ /datnica,  $e f; N2 1 \dots 191$ /svežanj, žnja  $m; N3 1 \dots 191$   
 ( $N1 > N2, N1 > N3$ )

$1 \dots 191$  – standardnojezični tekst iz oblasti informatike,  $Nn$  – podatak o čestoti  
 onda će datoteka biti izabrana kao ekvivalent jer ima veću čestotu.

Konačno, ako bi se pojavio takav slučaj gdje je čestota dvaju ekvivalenata identična, onda bi bilo primijenjeno »bacanje novčića« – nasumično izbacivanje jednog od mogućih ekvivalenata, što u binarnoj situaciji daje vjerojatnost pravilnog izbora od 0,5.

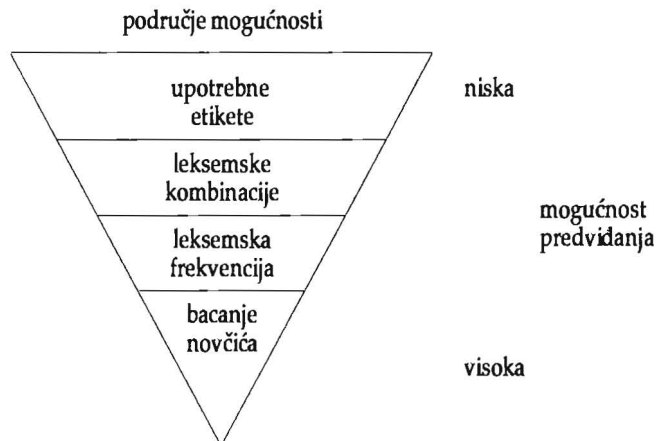
U osnovi, tijek procesa izbora leksičkog ekvivalenta ima "gusjeničnu" strukturu, koja je prikazana na crtežu:



$p$  – propušta se donošenje odluke  
 $p$  – odluka je donesena

Poslije svake donesene odluke korisnik ima mogućnost da je promijeni. Program zapisuje sve te interakcije i vodi statistiku o njima, što se onda koristi za korigiranje podataka.

Kako se moglo vidjeti iz predočenih primjera, princip rada programa takav je, da nakon svakog primijenjenog zahvata ostaje sve manje mogućih situacija s višestrukom ekvivalencijom. Drugim riječima, smanjuje se broj mogućnosti, a obrnuto razmjerno raste vjerojatnost pravilnog izbora, što je predočeno na ovom crtežu:



Kako se moglo vidjeti iz predočenih rješenja i primjera, to sužavanje mogućnosti i time podizanje vjerojatnoće pravilnog izbora ostvaruje se traženjem optimalnog odnosa između rješenja algoritamskog i heurističkog tipa.

## Literatura

- Fasold, Ralph. 1991. *Introduction to Sociolinguistics*. London : Basil Blackwell.  
Lipka, Leonhard. 1992. *An Introduction to English Lexicology*. Tübingen : Max Niemeyer.  
Šipka, Danko. 1994. Usage Labels Network: An Approach to Lexical Variation. *Linguistica (Ljubljana)* XXIV:2, 31–42.  
Zgusta, Ladislav. 1971. *Manual of Lexicography*. The Hague : Mouton.

## The algorithmic and heuristic processing of lexical data by the computer translator NeuroTran

### Summary

The authors explain how the computer translator program NeuroTran works, with some Croatian examples. The program works on the basis of minimal information grammar.

**Ključne riječi:** računalno prevođenje, program NeuroTran  
**Key words:** computer translating, NeuroTran program