

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Sašo Kuntarič

**Primerjava običajnih in faktorskih modelov pri statističnem
strojnem prevajanju iz angleščine v slovenščino z orodjem
Moses**

DIPLOMSKO DELO NA UNIVERZITETNEM ŠTUDIJU

Ljubljana, 2016

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Sašo Kuntarič

**Primerjava običajnih in faktorskih modelov pri statističnem
strojnem prevajanju iz angleščine v slovenščino z orodjem
Moses**

DIPLOMSKO DELO NA UNIVERZITETNEM ŠTUDIJU

MENTOR: prof. dr. Marko Robnik Šikonja
SOMENTOR: dr. Simon Krek

Ljubljana, 2016

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo: Statistično strojno prevajanje iz angleščine v slovenščino s sistemom Moses

Tematika naloge: Statistično strojno prevajanje je trenutno prevladujoč pristop k strojnemu prevajanju, ki na podlagi jezikovnih korpusov zgradi model jezika, ki ga nato uporablja za generiranje najverjetnejših prevodov. Eden uporabnejših odprtokodnih sistemov za statistično strojno prevajanje je Moses, ki za svoje delovanje potrebuje korpus vzporednih prevodov za dani jezikovni par. Sistem prilagodite za uporabo pri prevajanju iz angleškega v slovenski jezik. Na podlagi javno dostopnih besedilnih korpusov zgradite tudi ustrezen jezikovni model in preizkusite delovanje na zbirki računalniških besedil.

IZJAVA O AVTORSTVU
diplomskega dela

Spodaj podpisani Sašo Kuntarič sem avtor diplomskega dela z naslovom:

Primerjava običajnih in faktorskih modelov pri statističnem strojnem prevajanju iz angleščine v slovenščino z orodjem Moses

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Marka Robnika Šikonje in somentorstvom dr. Simona Kreka,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 22. 5. 2016

Podpis avtorja:

Zahvaljujem se prof. dr. Marku Robniku Šikonji za pomoč, izjemno odzivnost in potrpežljivost, dr. Simonu Kreku za nasvete in usmeritve ter prof. dr. Špeli Vintar za njene predloge pri izdelavi diplomske naloge.

Zahvaljujem se svojim staršem za finančno in moralno podporo ter odlične pogoje za študij. Zahvaljujem se ženi Nataši in hčerki Lani za spodbudo in razumevanje, ko sem namesto druženja z njima sedel za računalnikom.

Kazalo

POVZETEK

ABSTRACT

1. UVOD	1
2. STROJNO PREVAJANJE	3
2.1 Računalniško podprto prevajanje.....	3
2.2 Zgodovina strojnega prevajanja.....	4
2.3 Primeri uporabe strojnega prevajanja	5
2.4 Statistično strojno prevajanje.....	6
2.5 Statistično strojno prevajanje po besednih zvezah.....	9
2.6 Faktorski prevajalni modeli	10
2.7 Ocenjevanje strojnih prevodov	13
3. ORODJA IN TEHNOLOGIJE	17
3.1 SDLX in SDL Trados Studio 2015	17
3.2 m4loc.....	17
3.3 MXPOST	18
3.4 Obeliks	18
3.5 Strojni prevajalnik Moses	19
4. UČENJE IN STROJNO PREVAJANJE	21
4.1 Potek dela.....	21
4.2 Priprava korpusa	22
4.3 Ustvarjanje nefaktorskega modela.....	26
4.4 Ustvarjanje faktorskega modela.....	29
5. OCENJEVANJE IN ANALIZA PREVODOV	32
5.1 Analiza besedil.....	32
5.2 Ocenjevanje po metriki BLEU.....	33
5.3 Ocenjevanje s človeškim pregledom.....	35
6. SKLEPNE UGOTOVITVE IN PREDLOGI ZA NADALJNJE DELO	42
LITERATURA	45

Povzetek

Cilj diplomske naloge je prilagoditev sistema Moses za statistično strojno prevajanje iz angleščine v slovenščino. Strojno prevajanje je področje računalniške lingvistike, ki raziskuje uporabo programske opreme za prevajanje besedila iz enega jezika v drugega. Faktorsko statistično strojno prevajanje je podaljšek statističnega, pri katerem besedilu dodamo jezikovne oznake na ravni besed in jih spremenimo v vektorje. Tako želimo izboljšati kakovost dobljenih prevodov. Za odprtokodni prevajalnik Moses smo iz jezikovnega korpusa z besedili s področja informacijskih tehnologij ustvarili več faktorskih jezikovnih in prevajalnih modelov. Z njimi smo prevedli dve besedili s področja informacijskih tehnologij. Prvo je usmerjeno tržno in ima kompleksnejše zgradbo, drugo pa je bolj tehnične narave. Prevode, ki smo jih dobili, smo na dva načina primerjali med seboj ter z dvema neodvisnima človeškima prevodoma in s prevodom, ki smo ga ustvarili s storitvijo Google Translate. Za prvo primerjavo smo uporabili algoritem BLEU, za drugo pa so prevode pregledali človeški pregledovalci in podali subjektivno oceno, ki je pri prevajanju še vedno zelo pomembna. V zaključku smo si ogledali zanesljivost ocenjevalcev in analizirali rezultate ocenjevanja. Ugotovili smo, da so naši modeli primernejši za tehnična besedila, prehod na faktorske modele pa bolj vpliva na prevajanje kompleksnejših besedil.

Ključne besede: statistično strojno prevajanje, faktorsko strojno prevajanje, sistem Moses, jezikovni korpus, jezikovni model, prevajalni model, BLEU, evalvacija, Google Translate

Abstract

The aim of the thesis is to customise the Moses system for statistical machine translation from English to Slovenian. Machine translation is a field in computational linguistics that explores the use of software to translate text from one language to another. Factorised statistical translation is an extension of statistical machine translation, where language tags are added on the word level. Words are turned into vectors in an attempt to improve the translation quality. For the open-source machine translation system Moses we created multiple factorised language and translation models from a language corpus, containing IT-related texts. We translated two different IT-based documents. First one was marketing-orientated with a complex structure, while the second one was technical and straight-forward. We used two methods to compare the generated translations, two independent human translations and a translation, created by the Google Translate service. In the first comparison we used the algorithm BLEU and in the second comparison the translations were marked by human reviewers, who expressed a subjective score, which is very important in the translation field. In conclusion we calculated the inter-rater coherence and analysed the results. We discovered that our models were more suitable for technical texts, however switching to factorised models affects complex texts more.

Keywords: statistical machine translation, factorised machine translation, Moses system, language corpus, language model, translation model, BLEU, human evaluation, Google Translate

1. Uvod

Prevajanje in lokalizacija sta panogi, ki v zadnjih desetletjih doživljata velike spremembe. Iz tehnično relativno preprostega in počasnega prevajanja (odvisno od prevajalca) statičnih besedil se razvijata v računalniško podprti dinamični dejavnosti. Glavna razloga za to sta uvajanje orodij za računalniško podprto prevajanje in še mlajših strojnih prevajalnikov. Strojno prevajanje je postopek, pri katerem računalniški program analizira besedilo in brez posredovanja človeka ustvari prevedeno besedilo.

V spletu je na voljo veliko število orodij in korpusov za večje svetovne jezike, slovenščina pa je nekoliko manj podprta, kar je posledica naše majhnosti. Zaradi omejenega trga nismo dovolj zanimivi za učenje strojnih prevajalnih mehanizmov, kar pomeni, da moramo sami poskrbeti za ustrezno tehnološko podporo. Eden od razlogov za šibko podprtost in hkrati ovira za doseganje dobrih rezultatov je, da je slovenščina morfološko bogat jezik, zato je strojne prevajalnike težko naučiti slovnično pravilne slovenščine.

V diplomu želimo ugotoviti, kako dobra jezikovna in prevajalna modela je mogoče ustvariti in kako tematika uporabljenega korpusa vpliva na kakovost dobljenih prevodov. Odločili smo se za uporabo odprtokodnega mehanizma za strojno prevajanje Moses, ki omogoča učenje statističnih modelov prevajanja besedil iz izvornega v ciljni jezik. Nova besedila v izvornem jeziku s temi modeli dekodiramo in ustvarimo samodejne prevode v ciljni jezik. Učenje zahteva uporabo vzporednega korpusa besedila v dveh jezikih, ki ga običajno sestavljajo ročno prevedeni pari stavkov. Razvoj Mosesa je podprt v okviru projekta Connecting Europe Facility (CEF), ki ga financira Evropska komisija [4]. Zaradi prevladujočega mnenja o neprimernih slovenskih prevodih razširjene programske opreme smo se odločili, da bomo izdelali prevajalni model za besedilo, povezano s področjem računalništva. Dobljena prevoda bomo preverili na dva načina – s klasičnim človeškim pregledom in z računalniškim algoritmom BLEU. S prvim se bomo subjektivno prepričali, ali je prevod primeren za splošno uporabo in ali se približa prevodom, ki jih naredi profesionalni prevajalec, z drugim pa bomo dobili manj osebno, zato pa merljiv rezultat, kako dober je lahko prevajalni model. Preskusili bomo več modelov različnih kompleksnosti, od osnovnega nefaktorskega modela brez prerazporejanja besed v dobljenih prevodih do bolj zapletenih faktorskih modelov s prevajanjem, prerazporejanjem in ustvarjanjem novih besednih zvez v prevodu. Te modele bomo na zgoraj omenjena načina primerjali med seboj, nato pa še z dvema neodvisnima, človeško ustvarjenima prevodoma ter s prevodom, ustvarjenim s trenutno najbolj priljubljeno storitvijo za strojno prevajanje na tržišču, Google Translate. Tako bomo videli, ali se strojni prevodi že približajo človeškim in ali faktorski modeli izboljšajo dobljene prevode.

Diplomsko delo je razdeljeno na pet poglavij. V prvem se dotaknemo strojnega in računalniško podprtega prevajanja na splošno, nato pa podrobneje opišemo statistično strojno prevajanje, in sicer strojno prevajanje po besednih zvezah ter faktorsko prevajanje, ki ju omogoča sistem Moses. V tretjem poglavju predstavimo orodja in tehnologije, ki smo jih uporabljali (Moses, mxpost, Obeliks in druge), ter si ogledamo algoritem za evalvacijo strojnih prevodov BLEU. V četrtem poglavju opišemo svoje delo – namestitev programov, pripravo podatkov, ustvarjanje modelov in prevajanje besedila. Peto poglavje vsebuje povzetek narejenega ter pregled in analizo dobljenih rezultatov. V zadnjem poglavju predstavimo sklepne ugotovitve in predlagamo korake za nadaljnje delo.

2. Strojno prevajanje

Kot smo že omenili v uvodu, se področje prevajanja bliskovito spreminja. Roki za oddajo prevodov so vedno krajši, količina besedil za prevajanje pa hitro narašča. Zelo pomembno je dejstvo, da se v spletu pojavlja vedno več uporabniško ustvarjene vsebine (družabna omrežja, forumi itd.), za katero natančni in do potankosti pregledani prevodi niso potrebni, so predragi, za njihovo prevajanje pa ni na voljo dovolj časa, saj uporabniki te vsebine ustvarjajo neverjetno hitro. Uporabniku je v večini primerov dovolj, da približno razume smisel sporočila, popolna pravilnost in skladnost s slovnico pa mu nista tako pomembni. Vse to je vodilo k razmahu strojnega prevajanja, kjer bi računalnik naučili prevajati dovolj dobro, da bi obvladovali velike količine takšnih besedil, hkrati pa vseeno posredovali pomen sporočila uporabniku.

Strojno prevajanje, ki ga ne smemo zamenjevati z računalniško podprtim prevajanjem (razlike bomo opisali v naslednjem podpoglavju), je postopek, pri katerem računalniški program analizira besedilo in brez posredovanja človeka ustvari prevedeno besedilo [33]. Vloga uporabnika pri takih sistemih ni vselej povsem odpravljena, pri interaktivnih sistemih je za razreševanje večpomenskosti predvidena človekova pomoč, skoraj pri vseh sistemih pa je potrebna predpriprava izvirnega besedila in poprava prevedenega besedila [24].

2.1 Računalniško podprto prevajanje

Medtem, ko gre pri strojnem prevajanju za čim večjo avtomatizacijo prevajalskega procesa, so orodja za računalniško podprto prevajanje predvsem v pomoč in podporo prevajalcem. Poleg črkovalnikov, tezavrov, elektronskih slovarjev in drugih elektronskih podatkovnih virov gre pri računalniško podprtem prevajanju predvsem za programe za izdelavo in vzdrževanje terminoloških zbirk ter programe s pomnilniki prevodov. Dejanska uporabnost pomnilnikov prevodov je v veliki meri odvisna od značilnosti besedil, ki jih želimo prevajati, saj temelji na predpostavki, da se v določenih vrstah besedil stavki, besedni nizi ali posamezni izrazi v bolj ali manj nespremenjeni obliki ponavljajo. To velja predvsem za nekatere besedilne vrste iz družine tehničnih in strokovnih besedil, na primer navodila za uporabo, tehnične opise, nekatera pravna besedila itd. [24]

Pomnilnike prevodov omenjamo zato, ker predstavljajo tesno povezavo med strojnim in računalniško podprtim prevajanjem. Pogoj za uspešno statistično strojno prevajanje, ki je trenutno najbolj razširjeno, je namreč velika količina podatkov, predvsem dvojezičnih korpusov, ki jih dobimo ravno z uporabo zgoraj omenjenih orodij.

2.2 Zgodovina strojnega prevajanja

Kljub temu, da se je strojno prevajanje močno razširilo v zadnjih desetih letih, segajo njegovi začetki precej dlje v zgodovino. Že poskuse dešifriranja kod v 2. svetovni vojni lahko razumemo kot začetke strojnega prevajanja, saj lahko prevajanje tolmačimo kot prekodiranje enega niza znakov (besed) v drugega. Tako je eden od začetnikov strojnega prevajanja Warren Weaver dejal: »Ko pogledam članek v ruščini, pravim: "Besedilo je v resnici napisano v angleščini, vendar je zakodirano s čudnimi simboli. Dekodiral ga bom [27]."«

Po drugi svetovni vojni so s poskusi strojnega prevajanje prednjačili na univerzi MIT, že leta 1954 pa sta IBM in univerza Georgetown predstavili prvi sistem za strojno prevajanje – prevajal je iz ruščine v angleščino in vključeval besednjak iz 250 besed ter 6 slovničnih pravil. Prvi poskusi so se večinoma ukvarjali z neposrednim prevajanjem – zamenjavo besed na podlagi dvojezičnega leksikona. Vse skupaj je pod vprašaj postavilo poročilo organizacije ALPAC (Automatic Language Processing Advisory Committee), v katerem je bil leta 1966 zapisan zaključek, da strojno prevajanje s človeško popravo ni hitrejše ali cenejše od človeškega prevajanja in da je na voljo dovolj človeških prevajalcev. Kljub napačni predpostavki, da bi morale strojno prevajanje ustvarjati visokokakovostne prevode, je poročilo pokazalo na pomembno pomanjkljivost – previsoke obljube glede strojnega prevajanja.

Vseeno se raziskovanje ni popolnoma ustavilo in eden od najzgodnejših v celoti delujočih sistemov, ki je v uporabi še danes, je bil sistem za prevajanje vremenskih napovedi Meteo, ki je bil 1976 razvit na univerzi v Montrealu (iz tega lahko razberemo, da je lahko uspel le ozko usmerjen sistem, ki je imel omejen besednjak). 1968 je bil razvit sistem Systran, ki ga ameriško letalstvo uporablja od 1970, francosko-angleško različico pa je 1976 kupila Evropska komisija in na njeni osnovi razvila mehanizme za druge jezikovne kombinacije. Pri tem sistemu so že upoštevali pretekle izkušnje in prevajanje razdelili v tri faze – analizo, dvojezični prenos in sintezo. V istem času sta na tržišče prispela še dva sistema, Logos in METAL.

Nekoliko pozneje je poskusila Evropska gospodarska skupnost v projektu EUROTRA zaradi izrazite mnogojezičnosti Evropske unije s pristopom *interlingua*. Osnovna zamisel je bila prevajanje vseh jezikov v nekakšen medjezik oziroma univerzalno, jezikovno neodvisno strukturno in semantično ponazoritev stavčnih členov ter odnosov med njimi v odvisnosti od povedka. V svoji idealni različici bi taka vmesna oblika omogočala prevajanje v vse jezike sveta, saj bi bilo iz nje mogoče neposredno generirati poved v ciljnem jeziku, vendar se je kmalu izkazalo, da zadeva ni tako preprosta [24].

Ker je treba pri prevajanju sprejemati veliko število odločitev in upoštevati obsežno množico pravil, se je v 80-ih letih prejšnjega stoletja porodila zamisel, da bi za strojno prevajanje uporabili primere prejšnjih prevodov. Veliko raziskav prevajanja na osnovi primerov so opravili na Japonskem, konec 80-ih pa se je v IBM-ovih laboratorijih rodila zamisel o statističnem strojnem prevajanju, ki je bila posledica uspehov statističnih metod pri prepoznavanju govora. Žal je bila zamisel pred svojim časom, zato je večina strokovnjakov raziskave na tem področju opustila.

Raziskave so se v omejenem obsegu nadaljevale v 90-ih, pravi zagon pa so dobile po letu 2000. Temu je botrovalo več dejavnikov. 1998 so udeleženci delavnice na univerzi John Hopkins uspešno znova uporabili vse IBM-ove metode in dali orodja na voljo splošni javnosti. To je pritegnilo ameriško agencijo DARPA, da je financirala velika raziskovalna projekta TIDES in GALE. Zelo pomembno vlogo je odigral tudi ameriški odziv na teroristične napade 11. septembra 2001, saj se je močno povečalo zanimanje za prevajanje iz tujih jezikov, še posebej arabščine [18].

Zanimanje za statistično strojno prevajanje tako še nikoli ni bilo večje, saj uporabniki s sistemi, ki jih gostijo IBM, Microsoft in Google, na dan prevedejo več kot 50 milijonov spletnih strani. Tudi v sisteme, ki so na trgu že dlje časa, kot je Systran, ponudniki dodajajo statistične metode in jih tako izpopolnjujejo.

2.3 Primeri uporabe strojnega prevajanja

V grobem lahko opredelimo tri vrste sistemov glede na uporabo: tiste za objavljanje, razumevanje in komunikacijo [18]. Vsaka od teh uporab zahteva različno hitrost in kakovost strojnega prevajanja.

- **Objavljanje:**
 - popolnoma samodejno visokokakovostno strojno prevajanje, ki je zaenkrat udejanjeno le za omejene domene, kjer lahko iz preteklih primerov dobro določimo prihodnje prevode. Dobre rezultate lahko dosežemo tudi z nadzorovanim jezikom, kjer avtorji uporabljajo le določen besednjak.
 - poprava: strojno prevajanje je v tem primeru le prvi korak prevajanja, ki ga pregleda človeški prevajalec in tako zagotovi zeleno kakovost.
 - uporaba v prevajalskih orodjih: veliko prevajalcev strojno prevajanje še vedno sprejema skeptično. Da bi jim omogočili lažji prehod, lahko strojno prevajanje vdelamo v prevajalska orodja, kjer se prevajalec sam interaktivno odloči, ali bo sprejel ponujeni strojni prevod ali bo niz prevedel sam.

- Razumevanje:
 - grobo prevajanje: predstavlja trenutno najbolj razširjeno uporabo strojnega prevajanja. Gre za prevajanje spletnih strani, objav v družabnih omrežjih ter forumih in podobno. V tem primeru ni potrebno, da je prevod popoln, dovolj je, da ustrezno prenese pomen.
 - integracija z govornimi tehnologijami: na ta način lahko uporabniki strojno prevajajo telefonske pogovore ali radijske oddaje. Najbolj pomembna je seveda hitrost prevajanja. Uporablja se tudi pri bolj »senčnih« dejavnostih, kot sta vohunjenje in prisluškovanje.
- Komunikacija:
 - uporaba v ročnih napravah: skoraj zagotovo ste se v tujini že želeli pogovoriti z natakarjem, pa tega niste mogli, ker nista govorila istega jezika. Z uporabo strojnega prevajanja v ročni napravi vam je pri roki mehanizem, s katerim lahko izmenjate vsaj nekaj osnovnih besednih zvez ali stavkov in se tako sporazumete. Tak način uporabe bo postal nepogrešljiv v panogah, kot so turizem, zdravstvo in podobne.

2.4 Statistično strojno prevajanje

Statistično strojno prevajanje je vrsta strojnega prevajanja, ki temelji na analizi večje količine vzporednih besedil, iz katerih se s statističnimi algoritmi izračunavajo verjetnosti prevajalne ustreznosti za posamezne jezikovne možnosti [36].

Kot področje raziskovanja se je začelo v poznih 80-ih letih prejšnjega stoletja z IBM-ovim projektom Candide. Izvirni IBM-ov pristop preslikuje posamezne besede v besede in omogoča brisanje ter vstavljanje besed. Zamisel o statističnem strojnem prevajanju izhaja iz informacijske teorije. Dokument se prevede v skladu z verjetnostno porazdelitvijo $p(e|f)$, da je niz e v ciljnem jeziku prevod niza f v izvornem jeziku [12].

Eden od pristopov, ki se dobro obnese v računalništvu, je uporaba Bayesovega pravila:

$$\operatorname{argmax}_e p(e|f) = \operatorname{argmax}_e p(f|e)p(e) \quad (2.1)$$

Izraz (2.1) nam pove, da moramo za najboljši prevod poiskati najvišjo verjetnost $p(f|e)$. Prav tako nam pove, da lahko problem prevajanja razbijemo na dva ločena podproblema – ustvariti moramo jezikovni model $p(e)$ in ločen prevajalni model $p(f|e)$ [12].

Model temelji na modelu kanala s šumom, ki deluje na ta način: zamislimo si, da ima nekdo v glavi angleški stavek e . Ko ga natisnemo na papir, ga pokvari »šum« in tako postane slovenski stavek f . Da bi znova pridobili najbolj verjetni stavek e , moramo ugotoviti:

- kako ljudje govorijo v slovenščini (jezikovni model) in
- kako angleščino pretvarjamo v slovenščino (prevajalni model).

Najprej moramo ustvariti mehanizem, ki vsakemu slovenskemu stavku e dodeli verjetnost $p(e)$. Temu pravimo jezikovni model. Lahko ga poskusimo ustvariti tako, da mu dodamo čim več stavkov, vseeno pa se bodo vedno pojavljali taki, ki jih še nismo videli. Taki stavki dobijo $p(e) = 0$, tudi če so pravilni, kar seveda ni ustrezno. Iz tega razloga stavke razbijemo na dele. Če so deli ustrezni in se združujejo na pravilne načine, potem pravimo, da je niz slovenski. Za računalnik je stavke najlažje razdeliti na podnize, ki so lahko različnih dolžin. Podniz z n besedami se imenuje n -gram. Verjetnost b , da se bo beseda pojavila v trigramskem modelu, je torej:

$$b(z|xy) = \frac{\text{št. pojavitev}(\text{«}xyz\text{«})}{\text{št. pojavitev}(\text{«}xy\text{«})} \quad (2.2)$$

Povedano drugače, izraz (2.2) nam pove, kako verjetno je, da se bo beseda z pojavila, če sta pred njo že besedi x in y . Trigramski model si torej zapomni dve predhodni besedi. Bolj splošno: n -gramski model si zapomni $(n-1)$ predhodnih besed.

Tudi pri n -gramskih modelih lahko stavkom, ki jih model še ne pozna, dodelimo neničelne verjetnosti. Temu postopku pravimo glajenje. Razložimo ga lahko takole: če beseda » z « v našem besedilu še ni sledila nizu » xy «, se lahko vprašamo, ali je sledila besedi » y «. Če je bilo temu tako, je niz » xyz « morda ustrezen. Če tudi drugi pogoj ni bil izpolnjen, se lahko vprašamo, ali je » z « običajna beseda. Če tudi ta pogoj ni izpolnjen, bo niz » xyz « dobil zelo nizko verjetnost. Ob upoštevanju zgoraj navedenega se izraz (2.2) spremeni v:

$$b(z|xy) = 0,95 * \frac{\text{št. pojavitev}(\text{«}xyz\text{«})}{\text{št. pojavitev}(\text{«}xy\text{«})} + 0,04 * \frac{\text{št. pojavitev}(\text{«}yz\text{«})}{\text{št. pojavitev}(\text{«}z\text{«})} + 0,008 * \frac{\text{št. pojavitev}(\text{«}z\text{«})}{\text{skupno št. videnih besed}} + 0.002 \quad (2.3)$$

Koeficienti, navedeni v izrazu, so samo primer, in niso primerni za vse mehanizme. Običajno jih določimo s preskušanjem.

Kako torej vemo, kateri model je v našem primeru najboljši? Več modelov lahko med seboj primerjamo tako, da jih preskusimo na do takrat še nevidnih slovenskih preskusnih podatkih in se vprašamo, kakšna je verjetnost vsakega modela glede na preskusne podatke?

Zanima nas torej $p(\text{model}|\text{preskusni podatki})$. Če na tem izrazu uporabimo Bayesovo pravilo, dobimo:

$$p(\text{model}|\text{preskusni podatki}) = p(\text{model}) * \frac{p(\text{preskusni podatki}|\text{model})}{p(\text{podatki})} \quad (2.4)$$

Najboljši model je torej tisti, ki maksimizira verjetnost $p(\text{preskusni podatki}|\text{model})$.

V naslednjem koraku se moramo osredotočiti na $p(f|e)$, verjetnost slovenskega niza f , če imamo angleški niz e . Temu pravimo ustvarjanje prevajalnega modela. Išče torej e , ki maksimizira izraz $p(e) * p(f|e)$. To v našem primeru naredimo tako, da angleške besede zamenjamo s slovenskimi, ki jih na koncu premešamo [8]. Temu postopku pravimo IBM-ov model 3 (ta združuje leksikalni prevod, dodatni absolutni model poravnave in dodatni model rodovitnosti) [30]. IBM-ov model 3 je nadgradnje modela 1 in 2. Model 1 je zasnovan izključno na verjetnostni porazdelitvi leksikalnega prevajanja, ki je prevajanje besed v osamitvi. Iz danega korpusa model ugotovi, kateri prevod določene besede je najbolj verjeten in ga uporabi. Glavna slabost tega modela je, da obravnava vse vrste prerazporejanja besed kot enako verjetne. Model 2 prejšnjemu doda absolutni model poravnave. Na ta način je prevajanje sestavljeno iz dveh korakov – leksikalnega prevajanja in poravnave. Model 2 ustvari prevod tuje vhodne besede na položaju i s položajem prevedene besede na položaju j z verjetnostno porazdelitvijo poravnave. Do zdaj še nismo eksplicitno določili, v koliko besed se prevede vsaka vhodna beseda. V večini primerov se ena vhodna beseda prevede v eno izhodno, lahko pa se prevede v dve ali pa je v prevodu ni. Temu pravimo določanje rodovitnosti vhodnih besed, ki je podobno kot v zgornjih dveh primerih modelirana po verjetnosti porazdelitvi rodovitnosti.

Prednosti statističnega strojnega prevajanja:

- + lažje dostopni viri; veliko materiala v naravnem jeziku je dostopnega v digitalni obliki;
- + neodvisnost od jezikovnega para (četudi je statistično strojno prevajanje pri določenih jezikovnih parih ustrežnejše (slovenščina mednje ne spada));
- + manjši stroški kot pri metodah, ki temeljijo na pravilih; sistemi za strojno prevajanje na osnovi pravil zahtevajo ročno oblikovanje/razvijanje jezikovnih pravil, kar je časovno in finančno zahtevno; pravil prav tako pogosto ni mogoče posplošiti oziroma prenesti na druge jezike;
- + naravnejši prevodi; sistemi, ki temeljijo na pravilih, pogosto prevajajo dobesedno, statistično strojno prevajanje pa se tej težavi lahko (zaradi drugačnega pristopa) izogne;

Slabosti statističnega strojnega prevajanja

- poravnava povedi; v vzporednih korpusih je marsikatera poved izhodiščnega besedila prevedena v več povedi ciljnega besedila in obratno;
- prevajanje sestavljenk;
- prevajanje frazeologije;
- razlike v oblikoslovju različnih jezikov – oblikoslovne težave so precej pogoste pri slovenščini, saj je ta morfološko zelo bogata;

- razlike v besednem redu; različni jeziki imajo različen besedni red. Do določene mere se vrstni red besed lahko določi z običajnim vzorcem osebek–povedek–predmet (tako lahko govorimo o jezikih OPP (osebek–povedek–predmet), POP (povedek–osebek–predmet) ipd.). Dodatne razlike v besednem redu se pojavijo pri prilastkih ter med trdilnimi in vprašalnimi povedmi;

- besede zunaj besedišča (ang. *out of vocabulary (OOV)*); sistemi za statistično strojno prevajanje imajo v svojih bazah podatkov različne besedne oblike shranjene kot posebne simbole brez medsebojne povezave. Besednih oblik in stavkov, ki niso v bazi podatkov, ni mogoče prevesti – do tega pride zaradi pomanjkanja besedilnih virov, razlik v oblikoslovju različnih jezikov ipd. [36]

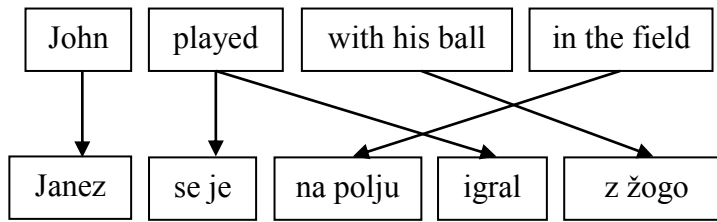
Statistično strojno prevajanje v grobem delimo na štiri vrste: prevajanje po besedah, prevajanje po besednih zvezah, prevajanje po slovnici in hierarhično prevajanje. V orodju Moses, ki ga bomo uporabljali za diplomsko delo, smo se odločili za prevajanje po besednih zvezah in njegovo razširitev, faktorsko prevajanje, ki si ju bomo ogledali podrobneje.

2.5 Statistično strojno prevajanje po besednih zvezah

V zadnjem času so raziskovalci dokazali, da je bolj kakovostne prevode kot s prevajanjem po besedah mogoče doseči z uporabo strojnega prevajanja po besednih zvezah [12]. Cilj takega prevajanja je zmanjšati omejitve prevajanja po besedah. Prednosti sta predvsem dve:

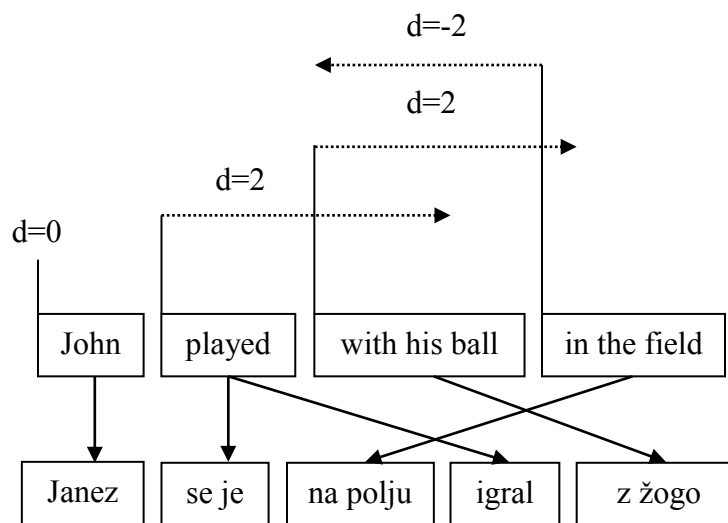
- Včasih se ena beseda v tujem jeziku prevede v več besed v ciljnem jeziku in obratno (na primer »played« – »se je igral«). Strojno prevajanje po besedah v takih primerih pogosto ne deluje pravilno.
- Prevajanje skupin besed nam pomaga, da razrešimo dvoumne prevode (beseda žvižgati ima na primer lahko (vsaj) dva pomena – »glasno je žvižgal pesem« ali »veter kar žvižga okoli ušes«).

Slika (2.1) prikazuje postopek prevajanja po besednih zvezah. Vhodni stavek se razdeli na nize zaporednih besed (ki se imenujejo besedne zveze). Vsaka angleška besedna zveza se prevede v slovensko, vrstni red dobljenih besednih zvez pa se lahko po prevajanju prerazporedi.



Slika 2.1: Pri prevajanju po besednih zvezah prevajamo besedne zveze in ne več posamezne besede.

Prerazporejanje se izvaja po modelu, ki temelji na razdalji (slika 2.2). To pomeni, da je premikanje besednih zvez na večje razdalje dražje kot premikanje na kratke razdalje.



Slika 2.2: Preurejanje besednih zvez po prevajanju in razdalje premikov.

Dokazano je bilo, da uporaba besednih zvez, daljših od treh besed, in učenje besednih zvez iz zelo natančnih sistemov za poravnavo besed na učinkovitost ne vplivata močno [17]. Zaradi tega smo se odločili, da uporabimo trigramski model.

2.6 Faktorski prevajalni modeli

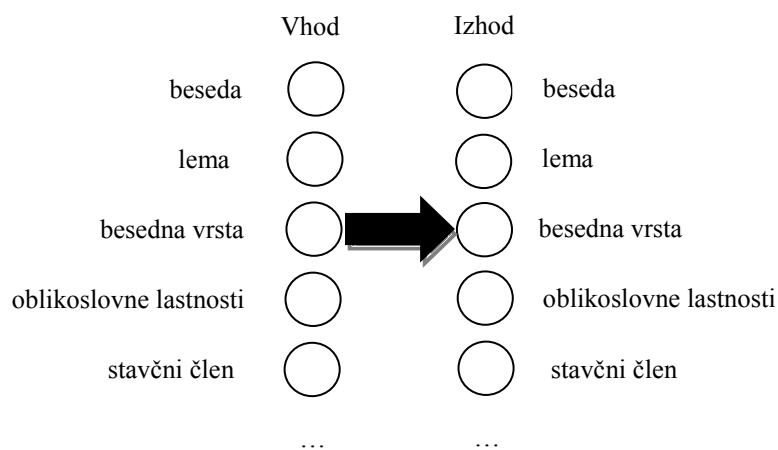
Trenutno najbolj razširjen pristop k statističnemu strojnemu prevajanju, t.i. prevajanje po besednih zvezah, je omejen na preslikovanje majhnih delov besedila (besednih zvez) brez uporabe dodatnih morfoloških, skladenjskih in semantičnih informacij. Dokazano je bilo, da je uporaba takih informacij v pred- ali poobdelavi koristna [13]. Preden se posvetimo faktorskim modelom, razložimo še pojma *morfologija* in *lema*, ki se bosta večkrat pojavila.

Morfologija je nauk o besednih vrstah, o oblikah posameznih besednih vrst in o njihovem naglasu ter o vlogah, ki jo posamezne besedne vrste in njihove oblike imajo v besedni zvezi in

v stavku [7]. V tem kontekstu je *morfem* najmanjša pomenska enota jezika (najmanjši del besed, ki ima samosvoj pomen). Morfološko bogati jeziki, med katere spada tudi slovenščina, imajo veliko kombinacij oblikoskladenjskih kategorij in lastnosti. Po specifikacijah, definiranih v okviru projekta Jezikoslovno označevanje slovenščine (JOS) [20], ima slovenščina takih kombinacij kar 1902, medtem ko jih ima angleščina kot izvorni jezik, iz katerega bomo prevajali, manj kot 100 (običajno okoli 60) [25].

Lema v morfologiji predstavlja kanonsko obliko nekega leksema. *Leksem* se v tem kontekstu nanaša na sklop vseh oblik neke besede, ki imajo enak pomen, lema pa se nanaša na besedo, ki je izbrana, da predstavlja leksem. V leksikografiji je to tudi izhodišče, citatna oblika oziroma jedro, po katerem je tudi indeksirana in prepoznavna. Leme imajo posebno vrednost v jezikih, kjer je veliko pregibanja, kot sta na primer češčina in slovenščina. Postopek določanja leme se imenuje *lematizacija* [37].

Faktorski prevajalni modeli dodajajo jezikovne oznake na ravni besed. Vsaka vrsta dodatnih informacij na ravni besede se imenuje *faktor* (slika 2.3). Ločen prevod leme in morfološki faktorji nam lahko pomagajo pri težavi s pomanjkanjem podatkov, poleg tega pa nam lahko dodatne informacije o stavčnih členih pomagajo pri preurejanju in zagotavljanju slovnične skladnosti. Prisotnost morfoloških značilnosti na ciljnih strani omogoča preverjanje ustreznosti samostalniških besednih zvez ali odnosov med predmetom in povedkom.



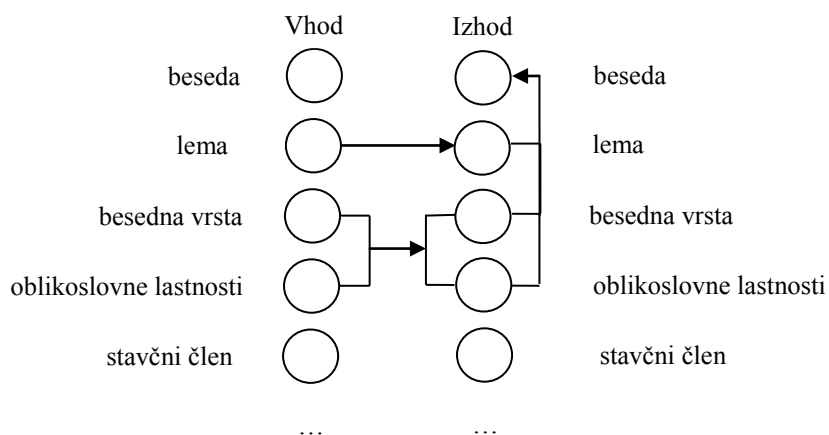
Slika 2.3: Faktorski prevajalni model, kjer so besede predstavljene kot vektorji faktorjev.

Pri običajnem statističnem prevajalnem modelu sta besedi hiša in hiše obravnavani kot popolnoma neodvisni besedi, kar je lahko pri morfološko bogatih jezikih težava, saj primerek besede hiša ne pomaga pri učenju prevoda besede hiše. V ekstremnih primerih bo prevajalni model poznal prevod besede hiša, prevod besede hiše pa mu bo popolnoma neznan, zato je prevajalni model za morfološke jezike smiselno ustvariti na podlagi lem, nato pa zbrati

podatke za različne oblike besed, ki izhajajo iz skupne leme. Če nadaljujemo s primerom besede hiše, se nam ta v faktorskem modelu spremeni v obliko: površinska oblika *hiše* | lema *hiša* | besedna vrsta *S* (samostalnik) | oblikoskladenjska oznaka *Sozer* (glede na oznake, določene v [25]).

Zaradi že omenjenega problema pomanjkanja podatkov je prevajanje besednih faktorjev smiselno razbiti na preslikovalne korake (slika 2.4):

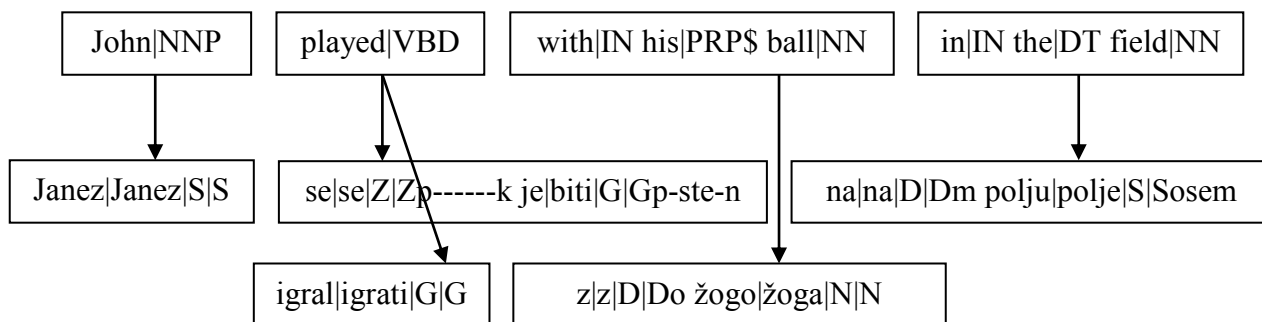
- prevajanje vhodnih lem v izhodne,
- prevajanje morfoloških in jezikovnih faktorjev ter
- ustvarjanje površinskih oblik glede na leme in jezikovne faktorje.



Slika 2.4: Prevajanje je razdeljeno v tri preslikovalne korake – prevod leme, prevod oblikoskladenjskih in morfoloških informacij ter ustvarjanje površinske oblike.

Učenje faktorskih modelov

Za učenje faktorskih modelov moramo običajnim vzporednim korpusom na vhodni in izhodni strani dodati faktorske oznake (slika 2.5). To običajno naredimo z avtomatiziranimi orodji, saj je ročno označevanje korpusov drago in počasno. Vsak preslikovalni korak ustvari del celotnega modela. To pomeni, da moramo pri učenju iz poravnanih korpusov ustvariti tabele za prevajanje in ustvarjanje ter določiti načine točkovanja za razlikovanje med dvoumnimi preslikavami.



Slika 2.5: Primer označenega prevoda. Izvirnik ima 2 dimenziji (površinsko obliko in oznako besedne vrste), prevod pa 4 (površinsko obliko, lemo, besedno vrsto in oblikoskladenjske informacije).

Dobljeni generacijski modeli so verjetnostne porazdelitve, ki se ocenijo samo na izhodni strani, pri tem pa lahko uporabimo dodatne enojezične podatke. Generacijski model se uči besedo po besedo.

Dekompozicija prevajanja besednih zvez v več korakov pomeni večjo računsko zahtevnost, saj je treba namesto po eni iskati po več tabelah in združevati njihovo vsebino.

2.7 Ocenjevanje strojnih prevodov

Pri strojnem prevajanju običajno delamo z velikimi korpusi besedil, zato bi bilo človeško pregledovanje preveč časovno potratno in predrago. Zaradi tega so se pojavile pobude, da bi se za to uporabljali računalniški algoritmi. Ti bi iz ocenjevanja odstranili subjektivnost (kar je hkrati dobro in slabo), predvsem pa bi zadevo pospešili in pocenili. Izbiramo lahko med več različnimi algoritmi, kot so NIST [35], METEOR [34] in LEPOR [32], mi pa smo se odločili za najpogosteje uporabljani BLEU.

BLEU (bilingual evaluation understudy) je algoritem za oceno kakovosti strojno prevedenega besedila iz enega naravnega jezika v drugega. Kakovost ocenjuje s primerjavo strojnega prevoda s človeškim. Osrednja ideja algoritma BLEU je: »Bolj podoben kot je strojni prevod človeškemu, boljši je [9].« Je ena od prvih metrik, ki je dosegla visoko korelacijo s človeško presojo kakovosti.

Vsak strojni prevod se primerja z enim ali več človeškimi (referenčnimi) prevodi istega stavka. Nemogoče je, da bi se taka stavka med seboj popolnoma ujemala, je pa zelo verjetno, da bo prevod z višjo stopnjo ujemanja bolj pravilen. Izziv za samodejno oceno prevoda je poiskati dobro mero za podobnost.

Začnemo lahko tako, da uvedemo natančnost (*precision*), ki je določena kot število besed, ki se pojavijo tako v strojnem prevodu kot v referenčnem besedilu:

$$\text{natančnost} = \frac{\text{št.pravilnih besed}}{\text{dolžina strojnega prevoda}} \quad (2.5)$$

Glavna težava pri natančnosti je, da ne upoštevamo besed, ki so v referenčnem prevodu, ni pa jih v strojnem. Zaradi tega uvedemo še priklic (*recall*):

$$\text{priklic} = \frac{\text{št.pravilnih besed}}{\text{dolžina referenčnega prevoda}} \quad (2.6)$$

Iz izrazov (2.5) in (2.6) je razvidno, da lahko obe meri prikrojimo. Če želimo zvišati natančnost, v prevodu uporabimo samo besede, za katere smo prepričani, da so pravilne, če pa želimo zvišati priklic, v strojnem prevodu uporabimo čim večje število besed in tako zagotovimo visoko število pravilnih.

Pri strojnih prevodih nas zanimata obe zgoraj omenjeni meri, zato jih združimo v mero F , ki je definirana kot harmonična sredina natančnosti in priklica:

$$F = \frac{\text{natančnost} \times \text{priklic}}{(\text{natančnost} + \text{priklic})/2} \quad (2.7)$$

Definirane mere lahko izračunamo za primer na sliki (2.1). Stavek *John played with his ball in the field* prevedemo kot *Janez se je na polju igral z žogo*. Ta prevod izberemo za referenčnega, hkrati pa ustvarimo dva strojna prevoda, za katera želimo izračunati natančnost, priklic in mero F .

1. strojni prevod: Janez je igrati na polju z žoga
Referenčni prevod: Janez se je na polju igral z žogo
2. strojni prevod: Na igrišču Janez se je igral z žogo

Natančnost prvega strojnega prevoda je 71 %, drugega pa 88 %. Priklic prvega prevoda je 63 %, drugega pa 88 % in mera F prvega prevoda je 67 %, drugega pa 88 %.

Kot vemo, pri prevodu niso pomembne samo besede, pač pa tudi njihov vrstni red. Za ocenjevanje le-tega uporabimo Levenshteinovo razdaljo. Ta je določena kot najmanjše število korakov urejanja, vstavljanj, brisanj in zamenjav, ki je potrebno, da sta dva stavka enaka. Z uporabo Levenshteinove razdalje lahko izračunamo stopnjo besednih napak (*word error rate*):

$$SBU = \frac{(\text{zamenjave} + \text{vstavljanja} + \text{brisanja})}{\text{dolžina referenčnega prevoda}} \quad (2.8)$$

Stopnja besednih napak je običajna mera učinkovitosti strojnih prevajalnikov. Z dinamično poravnavo nizov poravnamo izhodne stavke in referenčne prevode. Stopnja besednih napak normalizira število korakov urejanja glede na dolžino referenčnega prevoda.

Omejitev, da morajo biti vse besede v enakem vrstnem redu, je zelo stroga, zato je metrika BLEU prilagojena tako, da upošteva ujemanje daljših n-gramov (in ne samo besed) z referenčnim prevodom. Definirana je kot:

$$BLEU - n = kazen\ jedrnatosti \times \exp \sum_{i=1}^n \lambda_i \times \log natančnost_i \quad (2.9)$$

Kazen jedrnatosti znižuje oceno, če je prevod prekratek in tako zagotavlja, da ne poiščemo samo besed, za katere smo prepričani, da so pravilne. Definirana je kot:

$$kazen\ jedrnatosti = \min\left(1, \frac{\text{dolžina strojnega prevoda}}{\text{dolžina referenčnega prevoda}}\right) \quad (2.10)$$

Poglejmo naš primer:

1. strojni prevod: Janez je igrati na polju z žoga

1-gramski zadetek 2-gramski zadetek

Referenčni prevod: Janez se je na polju igral z žogo

2. strojni prevod: Na igrišču Janez se je igral z žogo

1-gramski zadetek 3-gramski zadetek

Če poznamo n-gramske zadetke, lahko izračunamo n-gramsko natančnost, to je razmerje pravih n-gramov za določen n glede na skupno število ustvarjenih n-gramov za ta n .

1. strojni prevod: 1-gramska natančnost 5/7, 2-gramska natančnost 1/6, 3-gramska natančnost 0/5 in 4-gramska natančnost 0/4.

2. strojni prevod: 1-gramska natančnost 7/8, 2-gramska natančnost 3/7, 3-gramska natančnost 1/6 in 4-gramska natančnost 0/5.

Tako lahko izračunamo BLEU-3 za drugi strojni prevod (če je katera koli n-gramska natančnost nič, bo BLEU za ta n nič), ki znaša 6,3 %.

Metrika BLEU ima podobno kot druge samodejne ocene veliko nasprotnikov, ki kot glavne slabosti navajajo:

- BLEU ne upošteva relativne pomembnosti posameznih besed. Nekatere besede so bolj pomembne od drugih. Primer je beseda *ne*, ki lahko zelo spremeni prevod, če jo izpustimo.
- BLEU deluje na lokalni ravni in ne upošteva splošne slovnične doslednosti. Izhodni prevod je lahko videti dobro, če upoštevamo samo n-grame, širša primernost pa je vprašljiva.

- Dejanske ocene BLEU so brez pomena. Nihče v resnici ne ve, kaj pomeni rezultat 30, saj je metrika sestavljena iz velikega števila dejavnikov, na primer števila referenčnih prevodov, jezikovnega para, domene in drugih.
- Nedavni izračuni t.i. človeških rezultatov BLEU, kjer so človeške prevode primerjali z referenčnimi človeškimi prevodi, so pokazali, da so bile ocene takih prevodov komajda (če so sploh bile) višje od strojnih prevodov, čeprav so bili sami prevodi veliko kvalitetnejši [18].

Vse zgoraj našteje slabosti pa ne veljajo samo za metriko BLEU. Vsaka metrika, ki uporablja primerjavo z enim ali majhnim številom referenčnih prevodov, ima namreč sistemsko težavo: v resničnem življenju lahko stavke prevajamo na več načinov. To pomeni, da je pristop s primerjavo, koliko se strojni prevod razlikuje od majhnega števila človeških, že v osnovi zgrešen.

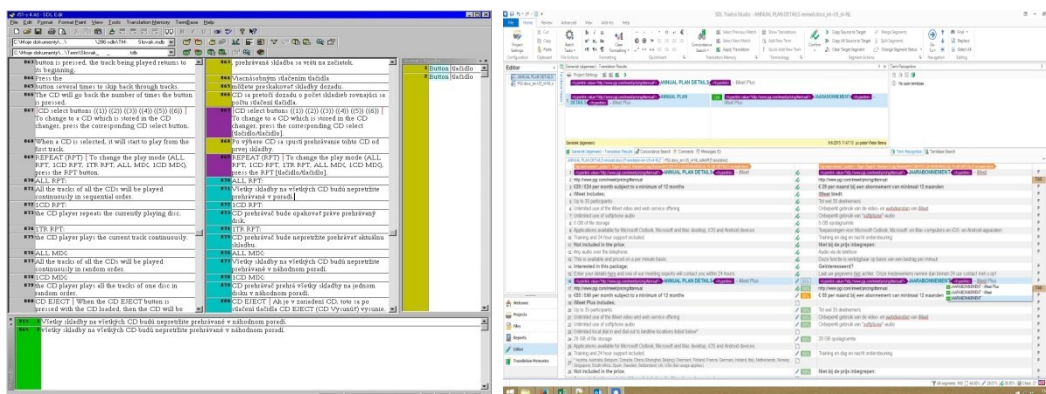
Za metriko BLEU smo se kljub pomanjkljivostim odločili, ker dosega visoko korelacijo s človeško oceno kakovosti, poleg tega pa je ena od najbolj priljubljenih samodejnih in nezahtevnih metrik.

3. Orodja in tehnologije

Preden začnemo z dejanskim učenjem in prevajanjem, si oglejmo orodja, ki jih bomo pri delu uporabljali.

3.1 SDLX in SDL Trados Studio 2015

Obe v naslovu omenjeni orodji sta programa za računalniško podprto prevajanje. Uporabljali smo ju, da smo pomnilnike prevodov, ki smo jih v prevajalski agenciji ustvarjali zadnjih deset let, pretvorili v izmenljivo obliko tmx, ki je primerna za nadaljnjo obdelavo in uporabo v strojnem prevajalniku Moses. Velika večina pomnilnikov je bila ustvarjenih v programu SDL Trados Studio, ker pa gre za daljše obdobje, jih je bilo nekaj tudi iz starejšega programa SDLX. Oba programa sta namenjena komercialni uporabi in nista prosto dostopna.



Slika 3.1: Programa SDLX in SDL Trados Studio 2015

3.2 m4loc

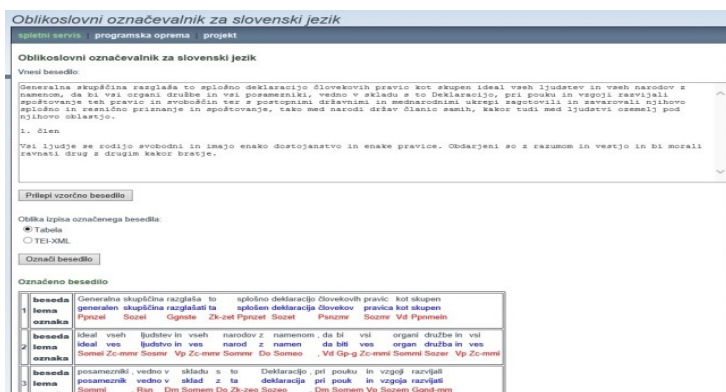
Orodje z nekoliko nenavadnim imenom je nastalo zaradi želje premostiti težave pri uporabi odprtokodnega strojnega prevajalnika Moses v prevajalski panogi. m4loc je zasnovan kot komplet orodij oziroma zbirka skript. To mu omogoča prilagodljivost pri integraciji v obstoječe lokalizacijske poteke dela [23], gre pa za odprtokodni projekt [21]. Pri našem delu smo uporabljali v jeziku Perl napisano skripto tmx2txt, ki dvojezični korpus v obliki zapisa tmx ekstrahira v dve ločeni besedilni datoteki v obliki zapisa txt.

3.3 MXPOST

Preden lahko začnemo ustvarjati faktorske modele, moramo korpusom dodati jezikovne oznake. Za ta namen je na voljo več označevalnikov, mi smo se pri angleščini odločili za program MXPOST, ki je tudi eden od predlogov na spletnem mestu strojnega prevajalnika Moses. Gre za javanski program, ki deluje na principu največje entropije. Napisal ga je Adwait Ratnaparkhi [1]. Program smo uporabili za označevanje angleškega korpusa, ki smo mu dodali informacije o besednih vrstah. Ker angleščina ni morfološko bogat jezik, je to za naše namene dovolj. MXPOST uporablja oznake, ki so bile določene v projektu Penn Treebank [19].

3.4 Obeliks

Za označevanje slovenskih korpusov smo uporabili program Obeliks, ki je izdelek Mihe Grčarja, Matjaža Juršiča, Jana Rupnika, Simona Kreka in Kaje Dobrovoljc [16]. Obeliks je računalniški program, s katerim poljubno besedilo razdelimo na enote in posameznim besedam pripišemo dodatne informacije, na primer besedno vrsto, v katero spada, njene lastnosti ali osnovno obliko, če gre za besedo, ki ima več pregibnih oblik. Označevalnik Obeliks (ob-likoslovni označ-e-valnik za s-lovenščino), ki je bil izdelan v okviru projekta Sporazumevanje v slovenskem jeziku [22], je sestavljen iz treh komponent: iz segmentacijskega in tokenizacijskega modula, ki besedilo razdeli na stavke ter besede, samega oblikoslovnega označevalnika, ki besedam pripiše besedno vrsto in njene lastnosti, ter lematizatorja, ki jim pripiše njihovo osnovno obliko (npr. delam -> delati, mizama -> miza). Več informacij in sama programska oprema so na voljo na [16]. Ker je slovenščina morfološko bogatejši jezik, smo površinskim oblikam v korpusu dodali še leme, besedne vrste in oblikoskladenjske informacije. Besede so se tako spremenile v vektorje s štirimi dimenzijami, ki smo jih uporabili za natančnejše strojno prevajanje.



Slika 3.2: Spletni vmesnik programa Obeliks.

3.5 Strojni prevajalnik Moses

Moses [14] je začel leta 2005 kot naslednika strojnega prevajalnika Pharaoh razvijati Hieu Hoang. Gre za implementacijo statističnega strojnega prevajanja, trenutno prevladujočega pristopa na tem področju.

Za učenje uporablja Moses vzporedne podatke (korpuse), za ugotavljanje ustreznih prevodov med dvema izbranim jezicoma pa uporablja skupne pojavitve besed in besednih zvez. V Moses je vključena tudi razširitev strojnega prevajanja po besednih zvezah, ki se imenuje faktorsko strojno prevajanje. Ta omogoča, da sistemu za prevajanje besednih zvez dodamo jezikovne informacije.

Moses je sestavljen iz dveh glavnih komponent: cevovoda za učenje in dekodirnika. *Cevovod za učenje* je zbirka orodij (večina je napisana v jeziku perl, nekaj pa v jeziku C++), ki iz vzporednih in enojezičnih neobdelanih podatkov ustvari modul za strojno prevajanje. Take podatke je treba pred uporabo pripraviti. Besedilo se razdeli na osnovne enote (*tokenization*) [3], te pa dobijo standardno veliko ali malo začetnico (*truecasing*). S hevrističnimi metodami se iz korpusov odstranijo pari, ki so videti nepravilno poravnani, in predolgi stavki. Vzporedni stavki se poravnajo po besedah, običajno z orodjem GIZA++ (vdelano v strojni prevajalnik Moses). Ta uporablja nabor statističnih modelov, ki jih je v 80-ih letih razvil IBM. Iz te poravnave Moses ekstrahira prevode besednih zvez, iz katerih se s statistiko celotnega korpusa ocenijo verjetnosti.

Pomemben del prevajalnega sistema je jezikovni model – statistični model, ki se ustvari iz enojezičnih podatkov v ciljnem jeziku. Dekodirnik skuša z njim zagotoviti, da je izhodni jezik tekoč. Za ustvarjanje jezikovnega modela uporablja Moses zunanja orodja, v našem primeru KenLM (več informacij je na voljo na strani avtorja Kennetha Heafielda [10]).

Zadnji korak v ustvarjanju prevajalnega sistema je prilagajanje (*tuning*), kjer se primerjajo različni statistični modeli za zagotavljanje najboljših prevodov. Privzeto je prilagajanje optimiranje rezultata BLEU (opisan je v prejšnjem razdelku) za prevedeni nabor za prilagajanje.

Drugi sestavni del Mosesa je *dekodirnik*. Njegova naloga je poiskati najboljše ocenjeni prevod v ciljnem jeziku (glede na prevajalni model), ki ustreza podanemu izvornemu stavku. Dekodirnik je napisan na modularni način in uporabniku omogoča, da ga prilagaja na spodaj opisane načine.

- Vhodni podatki: lahko so navadni stavki, besedilo z elementi, podobnimi tistim v jeziku xml, ali kompleksnejše strukture, kot so mreže.
- Prevajalni model: uporablja lahko pravila za prevajanje besednih zvez ali hierarhična pravila. Za hitrejšo nalaganje ga je mogoče pretvoriti v binarno obliko. Dopolniti ga je mogoče s funkcijami za dodajanje informacij postopku prevajanja, na primer funkcij, ki določajo vir parov besednih zvez za oceno zanesljivosti.
- Dekodirni algoritem: dekodiranje je velik iskalni problem, običajno prevelik za natančno iskanje. Moses uporablja različne strategije za tako iskanje, na primer iskanje po skladu (*stack-based search*), kubično odstranjevanje neželenih elementov (*cube-pruning*) ali razčlenjevanje grafikonov (*chart parsing*).
- Jezikovni model: Moses podpira več kompletov orodij za jezikovne modele (SRILM, KenLM, IRSTLM, RandLM), od katerih ima vsak svoje prednosti in slabosti [11].

Trenutno se Moses uporablja kot glavno orodje v projektu MosesCore, katerega cilj je spodbujanje razvoja in uporabe odprtokodnih orodij za strojno prevajanje. Projekt koordinira in financira Evropska unija [15].

4. Učenje in strojno prevajanje

4.1 Potek dela

Glavni namen naloge je raziskati zmožnosti statističnega strojnega prevajanja v sistemu Moses in narediti primerjavo med običajnimi ter faktorskimi modeli statističnega strojnega prevajanja.

Poskuse bomo izvajali na dveh sorodnih besedilih. Prvo navaja nekaj dobrih praks za fotografiranje in vključuje zgodbo fotografa, ki opisuje svoj vidik fotografiranja enega od večjih evropskih mest. Drugo besedilo je opis večnamenskih naprav na spletnem mestu enega od večjih proizvajalcev. Prvo besedilo je kompleksnejše in vsebuje nekaj prenesenih pomenov, s katerimi imajo strojni prevajalniki običajno težave, v drugem pa so stavki preprostejši, vključili pa smo tudi nekaj tehničnih lastnosti ene od naprav. Sklepamo, da bo prevajalnik bolje prevedel drugo besedilo, pri prvem pa se bo bolj poznal vpliv faktorkega prevajanja, saj je vrstni red besedila pomembnejši.

Izdelali smo naslednje modele strojnega prevajanja:

1. **Osnovni nefaktorski model:** začeli bomo z osnovnim nefaktorskim modelom, ki bo vseboval samo privzeto Mosesovo prerazporejanje besed glede na razdaljo. Pri tem modelu je premik besede za dve mesti dvakrat dražji kot premik za eno mesto.
2. **Nefaktorski model s prerazporejanjem:** prvi nefaktorski model bomo nadgradili z leksikalnim prerazporejanjem prevoda. Ta bo omogočal različne usmerjenosti, prerazporejanje v obe smeri in bo upošteval tako besedilo izvirnika kot prevoda.
3. **Osnovni faktorski model:** ustvarili bomo faktorski korpus, v katerem bomo površinski oblikam besed dodali oznake – besedno vrsto v izvorniku ter lemo, besedno vrsto in oblikoskladenjske informacije v prevodu. Pri prvem faktorskem modelu bomo površinsko obliko izvirnika prevedli v površinsko obliko in besedno vrsto.
4. **Faktorski model s prerazporejanjem:** pri četrtem modelu nas bo zanimalo, kako leksikalno prerazporejanje vpliva na faktorske modele, zato bomo 3. modelu dodali prerazporejanje po površinski obliki izvirnika in prevoda.
5. **Faktorski model s korakom ustvarjanja:** ta model bomo zasnovali tako, da bomo koraku prevajanja dodali še korak generiranja novih besed (*generation step*), s katerim bomo poskusili izboljšati kakovost dobljenih prevodov. Poleg tega izvornih površinskih oblik ne bomo prevajali v površinsko obliko, pač pa v lemo. Angleščina v nasprotju s slovenščino ni morfološko bogat jezik, zato so besede v imenovalniku

pogostejše, te pa je bolj smiselno prevajati v osnovno obliko slovenskih besed – leme. Prevedene besede bomo v koraku ustvarjanja povezali s površinsko obliko in besedno vrsto, da bi tako pridobili boljše prevode. Še vedno bomo uporabljali enak model leksikalnega prerazporejanja, le da bomo ta postopek izvajali glede na besedno vrsto izvirnika in prevoda.

6. **Faktorski model z dvema korakoma prevajanja:** kot zadnjega bomo ustvarili najkompleksnejši model, v katerem bomo ustvarili dve tabeli prevodov. To pomeni, da bomo imeli poleg koraka ustvarjanja in leksikalnega prerazporejanja dva koraka prevajanja. V prvem koraku bomo površinske oblike besed prevedli v leme, v koraku ustvarjanja leme povezali s površinskimi oblikami prevedenega besedila, nato pa besedne vrste izvirnika prevedli v besedne vrste prevoda. S tem želimo doseči višjo natančnost prevodov.
7. **Google Translate:** vse naše modele bomo primerjali s prevodom, ki smo ga ustvarili s storitvijo Google Translate. Ta omogoča brezplačno prevajanje krajših besedil. Naši besedili smo vnesli v storitev, ki nam je vrnila prevod. Tega smo razdelili na osnovne enote, kar je priporočeno pred uporabo Mosesove skripte za ocenjevanje po metriki BLEU.

Vse modele bomo primerjali med seboj ter z dvema neodvisnima prevodoma človeških prevajalcev. Za vse primerjane pare bomo izračunali podobnost BLEU, podmnožico posameznih prevedenih stavkov vsakega besedila pa bo z ocenami od 1 do 5 ocenilo tudi pet neodvisnih pregledovalcev, za katere bomo izračunali ujemanje. Tako želimo na dva načina, objektivnega in subjektivnega, oceniti prednosti in slabosti možnosti, ki jih ponuja Moses. Z metriko BLEU se bomo osredotočili na splošno oceno celotnega prevoda, s človeškim pregledom pa bomo podrobneje pogledali vsak stavek posebej.

4.2 Priprava korpusa

Pri statističnem strojnem prevajanju je zelo pomembna količina podatkov, ki jih imamo na voljo. Običajno velja – več primernega besedila imamo, boljši bodo strojni prevodi. Naš korpus je vseboval 2,4 milijona vrstic besedil s področja informacijskih tehnologij, ki so plod večletnega prevajanja v eni od prevajalskih agencij. Vsi prevodi so bili shranjeni v zbirkah prevodov programov SDLX in SDL Trados Studio. To je pomenilo, da smo se izognili koraku poravnave besedila (*alignment*), saj smo lahko poravnane pare besed izvozili v obliki tmx, ki je standard za izmenjavo take oblike besedila in ga podpira večina prevajalskih programov.

Nizi v obliki tmx so videti tako:

```
<tuv xml:lang="en-US">
<seg>Stream Internet radio, Pandora® and your music library with the touch of a
button.</seg>
</tuv>
<tuv xml:lang="sl-SI">
<seg>Z dotikom gumba pretakajte internetni radio, radio Pandora® in glasbeno
knjižnico.</seg>
</tuv>
</tu>
```

Preden lahko tako besedilo uporabimo v Mosesu, ga moramo pretvoriti, saj Moses kot vhodne podatke sprejema neoblikovano besedilo, izvornik in prevod pa morata biti v ločenih datotekah. Dodatna težava je tudi to, da zgoraj omenjeni prevajalski orodji delujeta v operacijskem sistemu Windows, Moses pa v operacijskem sistemu Linux. Zaradi tega smo za pretvorbo datoteke tmx prenesli v strežnik Linux, tam pa jih z orodjem m4loc pretvorili v ustrezne besedilne datoteke in jih združili v en korpus. Po pretvorbi so bili nizi videti tako:

Datoteka z izvornikom: *Stream Internet radio, Pandora® and your music library with the touch of a button.*

Datoteka s prevodom: *Z dotikom gumba pretakajte internetni radio, radio Pandora® in glasbeno knjižnico.*

V takih podatkih smo pred uporabo stavke razdelili na osnovne enote. Za latinične abecede to pomeni, da se med besede in ločila vstavijo presledki. Zgornji stavek je po tem postopku videti tako:

Datoteka z izvornikom: *Stream Internet radio , Pandora ® and your music library with the touch of a button .*

Datoteka s prevodom: *Z dotikom gumba pretakajte internetni radio , radio Pandora ® in glasbeno knjižnico .*

V naslednjem koraku smo besedam podali standardne majhne in velike začetnice. Lahko bi vse velike začetnice pretvorili v male, vendar nam na način, ki smo ga uporabili, ostanejo velike začetnice za imena in kraje:

Datoteka z izvornikom: *stream Internet radio , Pandora ® and your music library with the touch of a button .*

Datoteka s prevodom: *z dotikom gumba pretakajte internetni radio , radio Pandora ® in glasbeno knjižnico .*

Datoteki z izvornikom in prevodom moramo pred uporabo še očistiti. V tem postopku Moses iz korpusa odstrani prazne vrstice, ki lahko povzročijo nepravilno poravnavo nizov in s tem napačne prevode, ter vse stavke, ki so daljši od podane dolžine – ti lahko povzročijo težave pri poravnavi nizov.

Tak korpus je pripravljen za uporabo v nefaktorskih modelih, za faktorske modele pa je bilo treba datoteki z izvornikom in prevodom še oblikoskladenjsko označiti. Označevalniki besedila, ki so na voljo v spletu, ne sledijo skupnemu standardu, zato je pri vsakem potrebna dodatna obdelava besedila, s katero datoteke pretvorimo v obliko, ki jo lahko uporabimo v Mosesu.

Za označevanje angleških besedil smo uporabili program mxpost, ki je za uporabo precej preprost, besedilo pa nam vrne v obliki:

stream_NN Internet_NNP radio_NN , , Pandora_NNP ® , and_CC your_PRP\$ music_NN library_NN with_IN the_DT touch_NN of_IN a_DT button_NN . . .

Moses zahteva, da so faktorji med seboj ločeni z navpično črto (|), zato smo podčrtaje zamenjali z zahtevanim znakom. Ker se podčrtaj v besedilu skoraj ne pojavlja, smo lahko zamenjavo naredili kar ročno po celotnem korpusu. Gre za besedilo s področja informacijske tehnologije, zato smo morali biti previdni, saj lahko podčrtaji nastopajo v imenih spremenljivk, oznakah in podobnih elementih. Take elemente smo poiskali in nize, v katerih nastopajo, izbrisali iz korpusa. Končni rezultat so bili nizi v obliki

stream|NN Internet|NNP radio|NN , , Pandora|NNP ®| , and|CC your|PRP\$ music|NN library|NN with|IN the|DT touch|NN of|IN a|DT button|NN . . .

ki smo jih lahko uporabili v Mosesu. Mxpost doda površinski obliki besede le oznako besedne vrste, vendar je bilo to za naše poskuse dovolj.

Nekaj več dela smo imeli z označevanjem slovenske strani korpusa. Uporabili smo jezikoslovni označevalnik Obeliks, ki je pomnilniško zahteven. Korpus samo morali razdeliti na 5 delov in vzporedno obdelati vsakega posebej.

Obeliks besedilne datoteke pretvori v obliko XML-TEI, v kateri je bil naš stavek videti takole:

```
<p>
  <s>
    <w msd="Do" lemma="z">z</w>
    <S/>
    <w msd="Someo" lemma="dotik">dotikom</w>
    <S/>
    <w msd="Somer" lemma="gumb">gumba</w>
    <S/>
    <w msd="Ggnvdm" lemma="pretakati">pretakajte</w>
    <S/>
    <w msd="Ppnmeid" lemma="interneten">internetni</w>
    <S/>
    <w msd="Sometn" lemma="radio">radio</w>
    <S/>
    <c>,</c>
    <S/>
    <w msd="Somei" lemma="radio">radio</w>
    <S/>
    <w msd="Rsn" lemma="pandora">Pandora</w>
    <S/>
    <c>®</c>
    <S/>
    <w msd="Vp" lemma="in">in</w>
    <S/>
    <w msd="Ppnzet" lemma="glasben">glasbeno</w>
    <S/>
    <w msd="Sozet" lemma="knjižnica">knjižnico</w>
    <S/>
    <c>.</c>
  </s>
</p>
```

Za pretvorbo v obliko, ki jo podpira Moses, smo v skriptnem jeziku perl napisali skripto, ki nam omogoča, da v besedilno obliko pretvorimo poljubno število in kombinacijo faktorjev. Pri faktorskih modelih namreč potrebujemo jezikovni model, ki je sestavljen ne samo iz površinske oblike besed, pač pa tudi iz drugih faktorjev.

Če jo uporabimo v našem primeru, dobimo stavek v obliki:

```
z|z|D|Do dotikom|dotik|S|Someo gumba|gumb|S|Somer pretakajte|pretakati|G|Ggnvdm
internetni|interneten|P|Ppnmeid radio|radio|S|Sometn ,|,|, radio|radio|S|Somei
Pandora|pandora|R|Rsn in|in|V|Vp glasbeno|glasben|P|Ppnzet knjižnico|knjižnica|S|Sozet
.|.|.
```

Končna datoteka ima 4 faktorje – površinsko obliko, lemo, besedno vrsto in oblikoskladenjske informacije.

4.3 Ustvarjanje nefaktorskega modela

Jezikovni model

Ko imamo pripravljen korpus, lahko začnemo z ustvarjanjem jezikovnega modela. Z njim zagotovimo ne samo, da so besede pravilno prevedene, pač pa da je bolj pravilen tudi njihov vrstni red. Ustvarili bomo nefaktorski model, zato potrebujemo en jezikovni model, ki bo vseboval samo površinske oblike besed. V Mosesu ga ustvarimo z ukazom

```
~/mosesdecoder/bin/lmplz -o 3 < IT_corpus.sl > IT_corpus.arpa.sl.
```

Kot smo že omenili, bomo delali s 3-gramskim modelom, kar v ukazu navedemo s parametrom `-o 3`. Naključna vrstica v jezikovnem modelu je videti takole:

```
-2.0976682  izognite se ugibanju
```

Ker gre za velike datoteke (naš jezikovni model ima skoraj 9,5 milijona vrstic), jih pred uporabo za hitrejše delovanje in manjšo porabo pomnilnika binariziramo (*binarise*) – s tem dosežemo, da se v pomnilnik naložijo le besedne zveze, ki jih potrebujemo za določen stavek. V Mosesu to naredimo z ukazom

```
~/mosesdecoder/bin/build_binary IT_corpus.arpa.sl IT_corpus.blm.sl.
```

Prevajalni model

V naslednjem koraku ustvarimo prevajalni model:

```
~/mosesdecoder/scripts/training/train-model.perl -root-dir train -corpus IT_corpus_clean -f
en -e sl msd-bidirectional-fe -lm 0:3:$HOME/IT_corpus.blm.sl -external-bin-dir
~/mosesdecoder/tools -cores 32.
```

V njem Mosesu povemo, iz katerega korpusa naj ustvari model, kateri jezikovni model naj pri tem upošteva, na koncu pa še navedemo, da želimo delati z 32 jedri, s čimer močno skrajšamo ustvarjanje modela – v našem primeru je ustvarjanje prevajalnih modelov, tudi faktorskih,

trajalo okoli 9 ur. Uporabljali smo strežnik Dell PowerEdge M630 s 16 2,4 GHz procesorji Intel Xeon, 64 GB RAM-a in 64-bitnim operacijskim sistemom Linux Ubuntu.

Prevajalni model vedno vsebuje tabelo besednih zvez, izbirno tudi tabelo za leksikalno prerazporejanje, datoteko moses.ini, kjer so zapisane uteži za prevajalni model in druge podporne datoteke. Za nas sta najbolj zanimivi tabela besednih zvez in inicializacijska datoteka moses.ini. Če si ogledamo eno od vrstic v tabeli besednih zvez

```
!!! new password and confirmation ||| novo geslo in geslo za potrditev ||| 0.222222
1.21843e-14 1 0.00194155 ||| 3-0 4-1 5-2 4-3 6-5 ||| 9 2 2 ||| |||,
```

lahko vidimo, da je sestavljena iz izvorne ter prevedene besedne zveze, Mosesovega točkovanja, oznak za poravnavo besed in števil besed.

V datoteki moses. ini so poleg poti do datotek, ki jih Moses potrebuje za prevajanje, še uteži, ki pomagajo pri kakovosti prevoda (kazen za neznane besede, kazen za besede, kazen za besedne zveze ter uteži za prevajalni model, leksikalno preurejanje, če ga uporabljamo, popačenje in jezikovni model):

```
UnknownWordPenalty0= 1
WordPenalty0= -1
PhrasePenalty0= 0.2
TranslationModel0= 0.2 0.2 0.2 0.2
LexicalReordering0= 0.3 0.3 0.3 0.3 0.3 0.3
Distortion0= 0.3
LM0= 0.5
LM1= 0.5.
```

Poudariti je treba, da se vse te uteži ob ustvarjanju prevajalnega modela postavijo na privzete vrednosti, ki še niso primerne za prevajanje. Preden lahko model uporabimo za prevajanje, ga moramo prilagoditi.

Prilagajanje

Prilagajanje je postopek, s katerim želimo poiskati optimalne uteži za določen namen. To so tiste uteži, ki zagotovijo najboljši prevod majhnega nabora paralelnih stavkov – nabora za prilagajanje (*tuning set*). V našem primeru želimo zagotoviti, da bo prevod dosegel čim boljši rezultat po metriki BLEU. Moses ima v ta namen skripto mert-moses, ki opravi učenje za

doseganje najnižje ravni napake (*minimum rate training* [5]). Pri tem učenju Moses z danim prevajalnim modelom prevede izvirne besedne zveze in jih primerja z naborom referenčnih človeških prevodov, nato pa nastavitve prilagodi tako, da izboljša kakovost prevoda. Postopek opravi v več iteracijah – običajno od 10 do 15, lahko pa tudi več (v Mosesu je število iteracij privzeto omejeno na 25). Pri vsaki iteraciji se koraki ponovijo, dokler Moses ne najde prevoda najboljše kakovosti. Zakaj je prilagajanje potrebno? Kot smo opisali zgoraj, smo naš model sestavili iz jezikovnega in prevajalnega modela. Ta dva koraka in njuna izhoda sta neodvisna. Da bi dosegli višjo kakovost prevodov, ju je treba združiti, kar naredimo s prilagajanjem. Priporočljivo je, da je nabor za prilagajanje čim bolj podoben besedilu, ki ga želimo prevesti. V našem primeru smo uporabili 200-vrstični nabor, ki je podoben besedilu z opisi večnamenskih naprav. Zaradi tega je manj primeren za besedilo o fotografu, vendar se nismo odločili za drugačen nabor za prilagajanje, da bi si lahko ogledali tudi učinkovitost Mosesa v primeru, ko je nabor za prilagajanje nekoliko drugačen od besedila za prevajanje.

Prilagajanje v Mosesu zaženemo z ukazom

```
~/mosesdecoder/scripts/training/mert-moses.pl tuning_corpus.en tuning_corpus.sl
~/mosesdecoder/bin/moses moses.ini --mertdir ~/mosesdecoder/bin/ --decoder-flags="-
threads all".
```

V njem Mosesu navedemo izvorno in prevedeno stran nabora za prilagajanje ter datoteko *moses.ini*, v kateri so uteži, ki jih želimo prilagoditi. Izbirno lahko prilagajanje izvajamo z več nitmi, za kar ima skripta tudi priročen ukaz za uporabo vseh niti, ki jih strojna oprema ponuja (*--decoder-flags="-threads all"*). Po končanem prilagajanju se nam zgornje uteži spremenijo v:

```
LexicalReordering0= 0.0466654 0.0561246 0.103757 0.196091 -0.0178619 0.018535
Distortion0= 0.0404583
LM0= 0.0237383
LM1= 0.0277257
WordPenalty0= -0.191627
PhrasePenalty0= 0.0427445
TranslationModel0= 0.00820958 0.123724 0.0537373 0.0490006
UnknownWordPenalty0= 1.
```

Po končanem prilagajanju za hitrejše delovanje tabelo besednih zvez in tabelo za prerazporejanje binariziramo z ukazoma

```
~/mosesdecoder/bin/processPhraseTableMin -in phrase-table.gz -nscores 4 -out phrase-table
```

in

```
~/mosesdecoder/bin/processLexicalTableMin -in reordering-table.wbe-msd-bidirectional-
fe.gz -out reordering-table.
```

Naš sistem za statistično strojno prevajanje je pripravljen in želena datoteko lahko prevedemo z ukazom

```
~/mosesdecoder/bin/moses -f moses.ini < opisi_tiskalnikov.en > opisi_tiskalnikov.trans.sl.
```

4.4 Ustvarjanje faktorskega modela

Večina korakov za ustvarjanje faktorskega modela je podobna tistim za ustvarjanje nefaktorskega modela, zato bomo v tem podpoglavju izpostavili samo dodatne korake in razlike, ki jih zahteva ustvarjanje faktorskega modela.

Jezikovni model

Pri faktorskih modelih lahko ustvarimo jezikovni model za vsak faktor posebej, zato jih lahko hkrati uporabimo več. Skripto, omenjeno v poglavju 4.2, smo napisali tako, da s preprosto spremembo stavka

```
print OUT "$form|$lemma|$msd1|$msd1$msd2";
```

v

```
print OUT "$msd1";
```

izpišemo besedilno datoteko, v kateri je navedena samo besedna vrsta. Podobno lahko naredimo za poljubni faktor, nato pa z že omenjenim ukazom

```
~/mosesdecoder/bin/lmplz -o 3 < IT_corpus.sl > IT_corpus_POS.arpa.sl
```

ustvarimo jezikovni model, ki temelji na besedni vrsti. Vrstice v takem modelu so videti tako:

```
-0.8659348  G Z P.
```

V njih je prikazano, kakšna je verjetnost, da si v prevodu sledijo glagol, zaimек in pridevnik. Kot bomo videli pri ustvarjanju prevajalnega modela, lahko v enem modelu uporabimo tudi več jezikovnih modelov.

Prevajalni model

Ustvarjanje faktorskega prevajalnega modela zaženemo z ukazom

```
~/mosesdecoder/scripts/training/train-model.perl -root-dir test -corpus
IT_corpus_tagged_clean -f en -e sl -reordering msd-bidirectional-fe -lm
0:3:$HOME/IT_corpus_surface.blm.sl -lm 2:3:$HOME/IT_corpus_POS.blm.sl --translation-
factors 0-0,2 --generation-factors 0-1 --decoding-steps t0,g0 --reordering-factors 1-2 -
external-bin-dir ~/mosesdecoder/tools --cores 32.
```

Poleg parametrov, ki smo jih morali določiti pri nefaktorskem modelu, jih moramo v tem primeru dodati še nekaj. Kot smo že omenili, bomo uporabili dva jezikovna modela, enega na osnovi površinske oblike besed, drugega pa na osnovi besednih vrst. Paziti moramo, da uporabimo označen korpus, nato pa navedeno še parametre:

--translation-factors 0-0,2 – Mosesu povemo, katere faktorje na strani izvornika želimo prevesti v faktorje na strani prevoda. V našem primeru smo površinske oblike besed prevedli v površinske besede in besedne vrste.

--generation-factors 0-1 – navedemo, katere faktorje želimo v koraku ustvarjanja povezati med seboj. V našem primeru leme preslikamo v površinske oblike.

--decoding-steps t0,g0 – navedemo, v kakšnem vrstnem redu želimo opraviti korake. V tem primeru najprej opravimo korak prevoda, nato pa še korak ustvarjanja.

--reordering msd-bidirectional-fe in *reordering-factors 1-2* – podobno kot pri nefaktorskem modelu Mosesu povemo, da želimo izvajati prerazporejanje prevedenih besed, razlika je le v tem, da moramo navesti še, po katerem faktorju želimo prevod prerazporejati.

Razlika je tudi v tabeli besednih zvez, ki vsebuje faktorje. Naključni niz v našem primeru je videti tako:

```
2 % nDeny use over ||| %|% 2|K %|% nZavrni|G uporabo|S prek|D ||| 1 0.0200604 1
0.0124735 ||| 0-0 1-1 2-2 3-3 4-4 5-5 ||| 1 1 1 ||| |||
```


Ob elementih, ki so bili v datoteki pri nefaktorskem modelu, so v tabeli besednih zvez tudi dodatni faktorji, ki smo jih navedli v postopku ustvarjanja prevajalnega modela, zgoraj je to ob površinski obliki še besedna vrsta.

Prilagajanje

Za prilagajanje faktorkega modela smo uporabili podoben nabor za prilagajanje kot za prilagajanje nefaktorskega modela, glavna razlika je v tem, da mora biti izvorna stran nabora označena, prevedena stran pa ne.

5. Ocenjevanje in analiza prevodov

5.1 Analiza besedil

Za prevod smo si izbrali dve precej različni besedili, ki pa imata eno skupno lastnost – vsaka se na svoj način dotika področja informacijskih tehnologij. Prvo besedilo svetuje glede dobrih tehnik fotografiranja, v drugem delu pa predstavi priznanega fotografa, ki opisuje, kako je fotografiral Amsterdam. Besedilo je napisano v tržnem slogu, kar pomeni, da so stavki povezani, tekoči in da so v njih preneseni pomeni, na primer »*There is something truly magical about getting **up close and personal** with a subject.*«, kar na drugačen način pove, da se fotograf subjektu približa. S takimi strukturami imajo strojni prevajalniki običajno težave, zato nas je zanimalo, kako jih bo prevedel naš model. V besedilu je tudi nekaj za nas nenavadnih nizozemskih imen, na primer »*The iconic 17th century canals of **Herengracht, Prinsengracht, and Keizersgracht** stand out from the Centraal station in an almost semi-hexagonal shape.*«, ki lahko prav tako predstavljajo težavo za prevajalnik. Zaradi tega pričakujemo, da bo naš model za prevod tega besedila prejel nižjo oceno, da pa se bo bolj poznal prehod na faktorski model, saj je pri tekočem besedilu vrstni red besed še opaznejši kot pri tehničnih besedilih z več naštevanja.

Drugo besedilo je klasičen primer besedila, ki ga proizvajalci elektronskih naprav objavljajo na spletnih straneh. Sestavljeno je iz treh kratkih opisov večnamenskih naprav za optično branje, kopiranje, tiskanje in pošiljanje faksov. Poleg tega so v besedilu tudi tehnične lastnosti ene od naprav. Slog takega besedila je nekoliko bolj tog, stavki so krajši in bolj preprosti, v besedilu pa je več tehničnih izrazov, imen tehnologij, števil in podobno. Dobra primera stavkov v takih besedilih sta »*Wi-Fi and Wi-Fi Direct® connectivity ensure the flexibility every small business needs.*« in »*20 Pages/min Colour (plain paper 75 g/m²), 33 Pages/min Monochrome (plain paper 75 g/m²)*« (slednji je element seznama tehničnih lastnosti ene od naprav). Pričakujemo, da bo prevod takega besedila ocenjen bolje, vendar pa prehod na faktorsko prevajanje zaradi relativno preprostih stavčnih struktur ne bo tako občuten.

Odločili smo se za dolžino besedila, ki jo prevajalec v povprečju prevede v enem dnevu. Prvo besedilo je dolgo 2402 besedi, drugo pa 1437 besed.

5.2 Ocenjevanje po metriki BLEU

Moses ima za ocenjevanje po metriki BLEU namensko skripto *multi-bleu*, ki referenčno besedilo primerja z izbranim. Obe besedili sta lahko človeška ali strojna prevoda. Primerjavo izvedemo tako, da zaženemo ukaz

```
~/mosesdecoder/scripts/generic/multi-bleu.perl -lc referencno_besedilo.sl <
ocenjevano_besedilo.sl,
```

Moses pa nam rezultat vrne v obliki, kot je navedena v [9]:

BLEU = 23.47, 52.6/27.1/17.1/12.4 (BP=1.000, ratio=1.012, hyp_len=1866, ref_len=1843),

kjer je na prvem mestu globalni rezultat BLEU, za njim pa so 1-, 2-, 3- in 4-gramske natančnosti. V oklepaju sledijo kazni jedrnatosti, razmerje med dolžino izvornika in prevoda ter dolžini obeh besedil. V nalogi se bomo osredotočili na globalni rezultat BLEU.

Vse strojne prevode bomo primerjali z dvema neodvisnima človeškima prevodoma, zato nas je najprej zanimalo, kakšna je podobnost med obema prevodoma. Pri primerjavi obeh različic prvega besedila smo dobili rezultat 37,00, pri primerjavi drugega pa 46,90. Kot smo pričakovali, je drugi rezultat višji, saj so v prvem besedilu stavki kompleksnejši in jih je mogoče tolmačiti na več načinov, to pa pomeni, da je večja verjetnost, da ga bodo različni prevajalci prevedli drugače. Tehnična besedila morajo biti bolj natančna in pri njih imamo manj prevajalske svobode, zato je večja verjetnost, da bodo prevedena podobno. Kljub temu rezultata nazorno kažeta, kako različni so si človeški prevodi. Tudi pri precej nedvoumnem drugem besedilu obseg podobnosti ni presegel 50 %, zato se je odločitev, da vključimo tudi človeški pregled strojnih prevodov izkazala za smiselno in pravilno.

Ocenjevanje besedil

Rezultati ocenjevanja obeh besedil so zbrani v spodnjih tabelah:

	1. referenčni prevod	2. referenčni prevod
1. model	7,43	7,63
2. model	7,69	7,87
3. model	11,87	12,3
4. model	11,8	12,08
5. model	9,72	9,71
6. model	9,6	9,75
Google Translate	15,63	13,59

Tabela 5.1: Rezultati BLEU pri primerjavi strojnih prevodov prvega besedila z referenčnima.

	1. referenčni prevod	2. referenčni prevod
1. model	27,11	26
2. model	27,05	25,47
3. model	24,25	23,59
4. model	24,72	24,48
5. model	24,46	23,03
6. model	25,09	23,47
Google Translate	32,51	32,14

Tabela 5.2: Rezultati BLEU pri primerjavi strojnih prevodov drugega besedila z referenčnima.

Kot je razvidno iz rezultatov, se je večina naših domnev izkazala za pravilne. Pri prvem besedilu smo pri nefaktorskih modelih zaradi kompleksnosti besedila dobili nizke rezultate BLEU, ki jih tudi prerazporejanje prevoda ni bistveno izboljšalo (za 3,5 oz. 3 odstotke). Kot smo sklepali, je uporaba faktorskega korpusa naš rezultat bistveno izboljšala (za 33 oz. celo za 36 odstotkov). Ko smo modelu dodali prerazporejanje, se je rezultat nekoliko znižal (za 0,5 oz. za 2 odstotka). Še večji padec smo opazili, ko smo uporabili še korak ustvarjanja (18 oz. 21 odstotkov) in drugi korak prevajanja (19 oz. 21 odstotkov), vseeno pa je bil tudi najslabše ocenjeni faktorski model občutno boljši od najbolje ocenjenega nefaktorskega (za 20 oz. 19 odstotkov). Če rezultate primerjamo s tistimi, dobljenimi s prevodom iz storitve Google Translate, lahko ugotovimo, da smo se oceni kljub veliko manjšemu korpusu približali na 10 odstotkov, kar je zelo spodbuden podatek za nadaljnje delo s faktorskimi modeli.

Pri drugem besedilu je bil rezultat z osnovnim nefaktorskim modelom precej višji kot pri prvem besedilu. Znova smo ugotovili, da z dodajanjem prerazporejanja modela nismo izboljšali, saj se je rezultat za odtenek znižal. S prehodom na osnovni faktorski model se je rezultat znova znižal (za 10 oz. 9 odstotkov), vendar pa se je v tem primeru s prerazporejanjem znova nekoliko izboljšal (za 2 oz. 4 odstotke). Ko smo modelu dodali še korak ustvarjanja, je rezultat znova nekoliko padel (za 1 oz. 5 odstotkov) in se rahlo izboljšal, ko smo dodali še drugi korak prevajanja (za 2,5 oz. 2 odstotka). V primerjavi s storitvijo Google Translate je razlika tokrat nekoliko večja, saj je rezultat našega najboljšega modela za 17 odstotkov nižji kot pri prevodu s storitvijo Google Translate. Če pogledamo še razliko med najboljšim faktorskim modelom in storitvijo Google Translate, znaša ta 23 odstotkov.

Če med seboj primerjamo še rezultate istih modelov na obeh referenčnih besedilih, lahko ugotovimo, da so si izjemno podobni. V tabeli (5.1) vidimo, da je pri prvem besedilu največja razlika 3,5 odstotka in povprečna razlika 3 odstotke, tabela (5.2) pa prikazuje, da je pri drugem besedilu največja razlika 6,5 odstotka, povprečna razlika pa samo malo več kot 4 odstotke.

5.3 Ocenjevanje s človeškim pregledom

Pri drugem načinu ocenjevanja bomo del besedila dali v pregled petim neodvisnim človeškim pregledovalcem. Prosili jih bomo, da z oceno od 1 do 5 ocenijo vsak stavek posebej in napišejo komentar za posamezen model. Vsaka oseba bo pregledala 317 besed prvega in 301 besedo drugega besedila. Za vse modele bomo izračunali povprečno oceno in jo primerjali z rezultati BLEU, ki smo jih dobili v prejšnjem poglavju.

Da bi zagotovili čim bolj enoten kriterij, bomo vsem pregledovalcem posredovali enake kriterije za ocenjevanje. Po končanem pregledu bomo s povprečjem vseh parov Cohenovega koeficienta kapa [28] in s Fleissovim koeficientom kapa [29] preverili ujemanje ocen [31].

Pregledovalci bodo prevode ocenili po teh kriterijih:

- 5 - Stavek je popolnoma jasen in razumljiv. Prevod ni nujno popoln, vendar je slovnično pravilen, vse informacije pa so posredovane natančno.
- 4 - Stavek je v splošnem jasen in razumljiv. Ni popoln, je pa sprejemljiv, mogoče ga je razumeti in zajame večino pomena izvirnika.
- 3 - V stavku so slovnične napake in/ali napačno izbrane besede. Z nekaj truda je mogoče razbrati del, ne pa vsega, pomena izvirnika.
- 2 - V stavku je nekaj ključnih besed, vendar je v prevodu izraženega malo pomena izvirnika.
- 1 - Prevod je nesprejemljiv. Stavek je nerazumljiv, natančno je posredovano malo ali nič informacij. Prevod ne odraža pomena izvirnika.

Rezultati pregledovanja so zbrani v spodnjih tabelah:

1. besedilo

	1. pregledovalec	2. pregledovalec	3. pregledovalec	4. pregledovalec	5. pregledovalec
1. model	2,0434	1,5217	1,3043	1,6521	1,5217
2. model	2	1,7391	1,3913	1,6522	1,6364
3. model	2,4783	2,6522	2,2609	2,3478	2,3913
4. model	2,4783	2,4782	2,3478	2,1739	2,3913
5. model	2,3478	2,4348	2,1739	2,1304	2,1739
6. model	2,261	2,4783	2,0434	1,7826	2
Google Translate	3,1739	3,5217	3,0435	3,1304	3,2174

Tabela 5.3: Ocene, ki so jih pregledovalci dodelili modelom za prvo besedilo.

2. besedilo

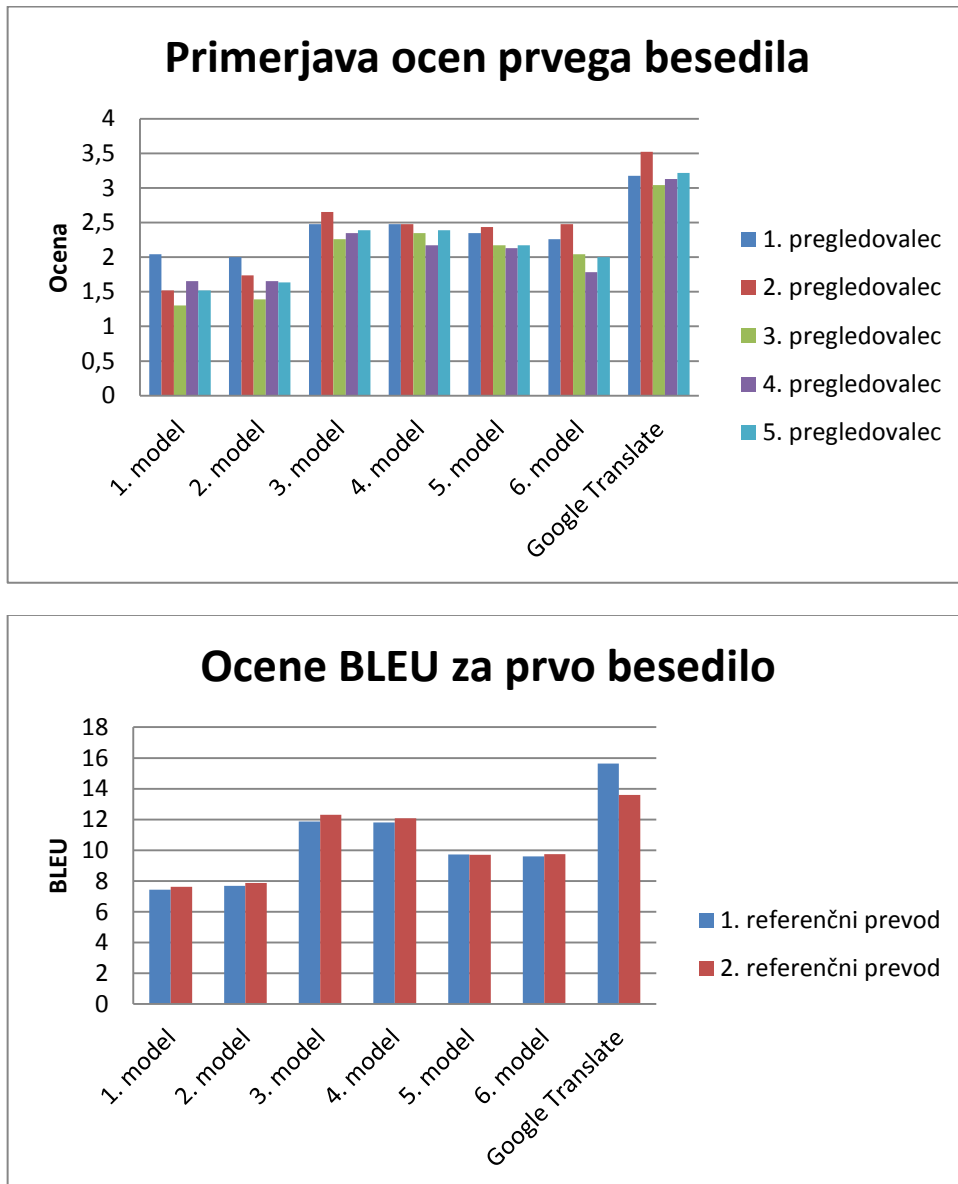
	1. pregledovalec	2. pregledovalec	3. pregledovalec	4. pregledovalec	5. pregledovalec
1. model	3,06	2,66	2,52	2,551	2,4898
2. model	2,8571	2,36	2,22	2,24	2,26
3. model	3,2041	2,94	2,8	2,92	2,84
4. model	3,42	3,22	3,02	3,2	3,16
5. model	3,3673	3,22	3,06	3,24	3,1
6. model	3,4	3,32	3,12	3,04	3,1
Google Translate	3,86	3,86	3,4	4,0204	3,6

Tabela 5.3: Ocene, ki so jih pregledovalci dodelili modelom za drugo besedilo.

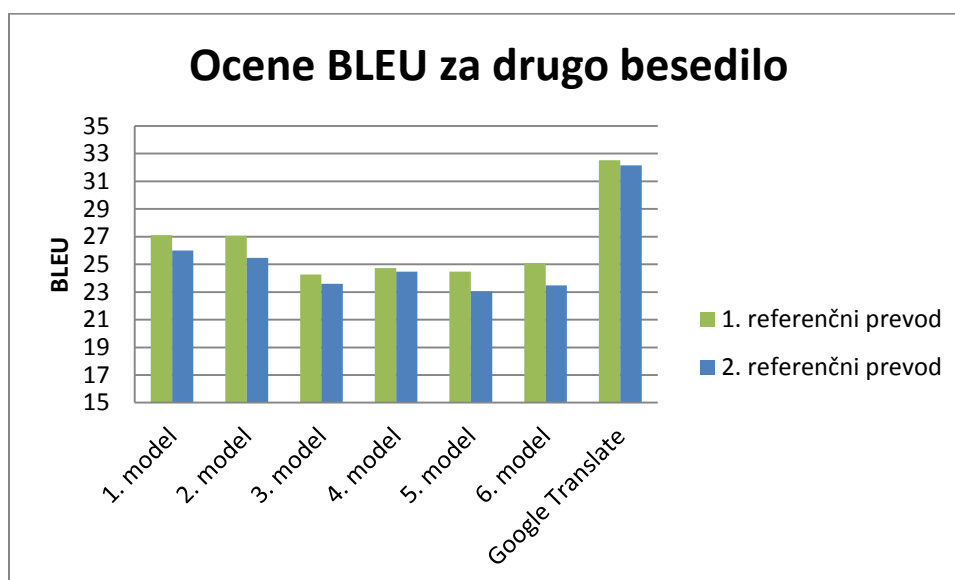
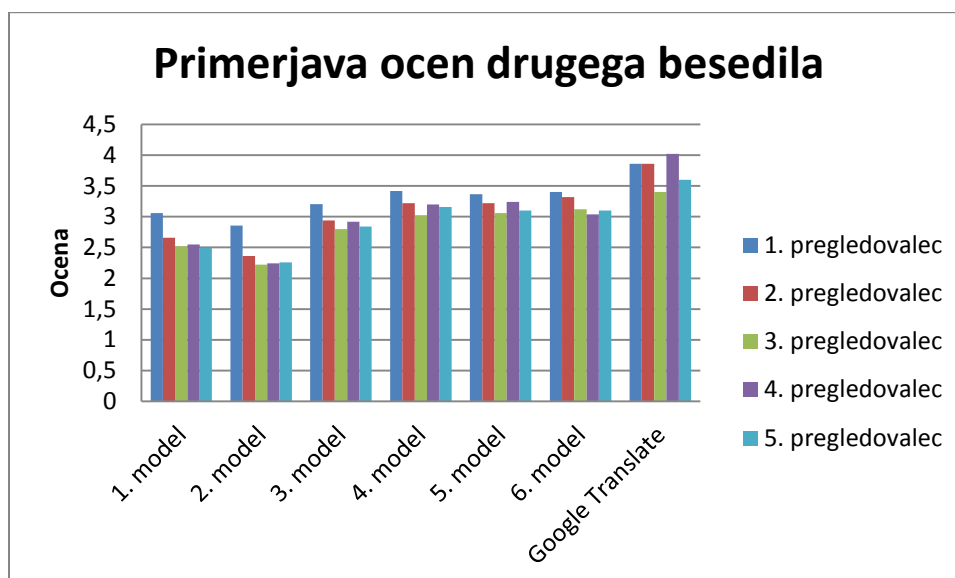
Pregledovalci so si bili v mnenju precej enotni in so potrdili naše domneve. Pri prvem besedilu je bil nefaktorski model (s prerazporejanjem ali brez njega) precej neuporaben, saj je poleg slabih prevodov tudi velik del besedila pustil nepreveden. Pri faktorskih modelih povprečna ocena precej poskoči in od modela do modela niha. Nekateri pregledovalci so s prevodi do določene mere zadovoljni, drugi še vedno menijo, da prevod ni uporaben. Vsi se strinjajo, da je prevod s storitvijo Google Translate najboljši.

Pri drugem besedilu pregledovalci ugotavljajo, da je prevod že pri nefaktorskih modelih boljši, vendar pa se ta kakovost z uporabo faktorskih modelov ne poveča. Tudi v tem primeru je Google Translate delo opravil najboljše.

Če grafa z ocenami, pridobljenimi z algoritmom BLEU, primerjamo z ocenami pregledovalcev, dobimo spodnji sliki:



Graf 5.8: Primerjava ocen človeških pregledovalcev in ocen BLEU za prvo besedilo.



Graf 5.9: Primerjava ocen človeških pregledovalcev in ocen BLEU za drugo besedilo.

Čeprav rezultatov ne moremo primerjati neposredno zaradi različnih metrik, na grafih (5.8) in (5.9) jasno vidimo, da se gibanje ocen kakovosti pri obeh besedilih dobro ujema. Pri prvem besedilu so rezultati z nefaktorskimi modeli nizki, pri prehodu na faktorske modele se občutno izboljšajo, pri različnih faktorskih modelih pa nekoliko nihajo.

Pri drugem besedilu so rezultati že z nefaktorskimi modeli boljši, vendar pa pri prehodu na faktorske modele pride do edine občutne razlike med računalniško in človeško oceno – ocena BLEU je za faktorske modele nižja, človeški pregledovalci pa so jih ocenili bolje kot nefaktorske.

Po pregledu vseh rezultatov bomo preverili še stopnjo strinjanja med ocenjevalci. Ta ocenjuje, koliko homogenosti ali strinjanja je v ocenah ocenjevalcev. Uporabna je pri izboljševanju orodij, danih človeškim ocenjevalcem. Če se različni ocenjevalci ne strinjajo, je metrika slaba ali pa je treba ocenjevalce bolje podučiti. Za ugotavljanje stopnje strinjanja je na voljo več

statistik, ki so odvisne od vrste meritev. V našem primeru smo uporabili Cohenov koeficient kapa in Fleissov koeficient kapa.

Cohenov koeficient kapa je statistična mera, ki določa strinjanje med ocenjevalci za kvalitativne elemente. V splošnem velja za robustnejšo mero kot preprost izračun odstotka ujemanja, saj upošteva tudi strinjanje, do katerega pride naključno. Koeficient meri strinjanje med dvema ocenjevalcema, ki N elementov razvrstita v C medsebojno izključujočih se kategorij. Cohenov koeficient kapa izračunamo kot

$$\lambda = \frac{p_o - p_e}{1 - p_e} = \frac{1 - (1 - p_o)}{1 - p_e}, \quad (5.1)$$

kjer je p_o relativno opazovano strinjanje med ocenjevalci, p_e pa je hipotetična verjetnost naključnega strinjanja, kjer se iz opazovanih podatkov izračunajo verjetnosti, da vsak opazovalec naključno izbere eno kategorijo. Če se ocenjevalci popolnoma strinjajo, potem velja $\kappa = 1$. Če med ocenjevalci ni drugega strinjanja kot naključno, velja $\kappa \leq 0$.

V našem primeru smo Cohenov koeficient kapa izračunali s kompletom statističnih orodij Real Statistics [2] za Excel. Najprej smo ga izračunali za vsak par pregledovalcev vseh modelov, nato pa smo določili povprečni koeficient najprej za vsak model, nato pa še za vse modele skupaj. Matrika ujemanj za vsak par pregledovalcev je videti tako:

1. pregledovalec	2. pregledovalec				
	1	2	3	4	5
1	6	1			
2	16	7	3		
3	5	10	10		
4		1		4	4
5				2	4

Tabela 5.8: Matrika ujemanj za izračun Cohenovega koeficienta kapa za vsak par pregledovalcev.

V prvem stolpcu tabele (5.8) so ocene, ki jih je posameznim stavkom dodelil eden od pregledovalcev, v drugi vrstici pa ocene drugega pregledovalca (od 1 do 5). Po diagonali matrike so navedene ocene, pri katerih sta se pregledovalca strinjala, nad diagonalo in pod njo pa tiste, pri katerih se nista.

	1. model	2. model	3. model	4. model	5. model	6. model	7. model
Povprečni Cohenov koeficient kapa	0,372	0,3544	0,4127	0,3893	0,4343	0,3707	0,4122
	Povprečni Cohenov koeficient kapa vseh modelov:						0,3922

Tabela 5.9: Povprečni Cohenov koeficient kapa za vse modele.

V tabeli (5.9) lahko vidimo, da so si bili pregledovalci najbolj enotni pri 5. modelu, najmanj pa pri 2. modelu. Po lestvici, ki sta jo predlagala Landis in Koch [6], je naš povprečni Cohenov koeficient kapa ravno na meji med dokajšnjim in zmernim strinjanjem, kar je glede na subjektivno naravo človeškega pregledovanja pričakovan in sprejemljiv rezultat.

Cohenov koeficient kapa meri le strinjanje med dvema ocenjevalcema. Za več ocenjevalcev uporabimo Fleissov koeficient kapa, ki je posplošitev Scottove statistike p_i in ne Cohenove kape. Tudi ta koeficient bomo izračunali s kompletom statističnih orodij Real Statistics za Excel.

Fleissov koeficient kapa je statistična mera za določanje zanesljivosti strinjanja med nespremenljivim številom ocenjevalcev pri dodeljevanju kategoričnih ocen množici elementov ali elementov za razvrščanje. Mera izračuna stopnjo strinjanja v razvrstitvi in jo primerja s tisto, ki bi bila pričakovana pri naključnem razvrščanju. Izračunamo jo po izrazu

$$p_a = \bar{p} = \frac{1}{n} \sum_{i=1}^n p_i = \frac{1}{n} \sum_{i=1}^n \frac{\sum_{j=1}^k x_{ij}^2 - m}{m(m-1)} = \frac{1}{mn(m-1)} \left[\sum_{i=1}^n \sum_{j=1}^k x_{ij}^2 - mn \right], \quad (5.2)$$

kjer je n število subjektov, k število ocenjevalnih kategorij in m število ocenjevalcev vsakega subjekta. V našem primeru je 5 pregledovalcev ocenjevalo 73 stavkov, ki jih je dodeljevalo 5 kategorijam (ocenam od 1 do 5). Matrika za vsak model je bila videti tako:

	1	2	3	4	5
1	2	3			
2	1	3	1		
3	1	4			
4	4	1			
5	2	2	1		
6	3	1	1		
7	4	1			
8	4	1			
9	1	3		1	
10	2	3			
...

Tabela 5.10: Matrika za izračun Fleissovega koeficienta kapa za vsak model (zaradi preglednosti smo prikazali samo prvih deset stavkov).

V prvem stolpcu tabele (5.10) so navedeni posamezni stavki, v prvi vrstici pa ocene od 1 do 5. V polje pod vsako oceno vnesemo število pregledovalcev, ki je stavku dodelilo to oceno.

	1. model	2. model	3. model	4. model	5. model	6. model	7. model
Povprečni Fleissov koeficient kapa	0,3613	0,3598	0,4222	0,391	0,427	0,3682	0,4199
	Povprečni Fleissov koeficient kapa vseh modelov:						0,3928

Tabela 5.11: Povprečni Fleissov koeficient kapa za vse modele.

Iz tabele (5.11) lahko vidimo, da sta oba koeficienta ujemanja podobna. Tudi glede na Fleissov koeficient kapa so se pregledovalci najbolj strinjali pri 5. modelu, najmanj pa pri 2. modelu. Fleissov povprečni koeficient je zelo podoben Cohenovemu in je na meji med dokajšnjim ter zmernim strinjanjem.

6. Sklepne ugotovitve in predlogi za nadaljnje delo

V diplomski nalogi smo poskušali odgovoriti na dve glavni vprašanji: je strojno prevajanje že primerno za splošno rabo in ali faktorski modeli izboljšajo kakovost strojnega prevoda? Strnimo naše ugotovitve. Strojni prevajalni modeli se dobro obnesejo pri prevodih tehnične narave, to so besedila, kjer so stavki relativno preprosti, poleg tega pa niso dvoumni in vsebujejo malo prenesenih pomenov. Pregledovalci so prevode takih stavkov pri večini modelov ocenili s 4 ali 5, kar pomeni, da pomena takega stavka ni težko razumeti. Drugačno sliko dobimo pri besedilih, ki so bolj zapletena. Strojni prevajalniki imajo z daljšimi in kompleksnimi stavki še vedno veliko težav. To ne velja samo za modele, ki smo jih ustvarili sami, pač pa tudi za Google Translate, ki se ponaša z veliko večjim korpusom kot naši modeli. To je dobro vidno ne samo iz subjektivnega pregleda, pač pa tudi iz rezultatov BLEU, ki so v našem primeru pri bolj zapletenem besedilu nižji tudi za več kot polovico. Trdimo lahko, da strojno prevajanje dobro služi namenu prenosa približnega pomena v drug jezik, ni pa še primerno za visokokakovostne prevode brez človeškega pregleda (izjema so seveda močno specializirana besedila, kjer je besedišče veliko ožje).

Kako na strojne prevode vplivajo faktorski modeli? Pri tehničnem besedilu uvedba faktorskega modela ni imela občutnega vpliva. Rezultat BLEU je bil celo nekoliko nižji kot tisti z nefaktorskimi modeli, človeška ocena faktorskih strojnih prevodov je sicer nekoliko višja, vendar ne dovolj, da bi upravičila čas in sredstva, ki jih je treba vložiti v ustvarjanje faktorskih korpusov ter modelov. Drugačne rezultate smo dobili pri prevajanju kompleksnejšega besedila. Tam sta računalniška in človeška ocena pri uvedbi faktorskega modela močno poskočili in se povečali skoraj za polovico. V najboljšem primeru se je storitvi Google Translate ocena BLEU približala na 12 odstotkov, kar je glede na neprimerno manjši korpus odličen rezultat in velika spodbuda za nadaljnje delo s faktorskimi modeli. Tako je na primer ocena prevoda niza »*Street in Amsterdam*« pri enem od pregledovalcev z ocene 1 zrasla na 5 (iz »*Street v Amsterdam*« v »*Ulica v Amsterdamu*«) in ocena prevoda »*They say nothing lasts forever.*« z 1 na 3 (iz »*They izgovorite nič zdrži forever.*« v »*Te izgovorite nič ne traja za vedno.*«). Tudi v svojih komentarjih so pregledovalci zapisali, da je prevod prvega besedila z uvedbo faktorskih modelov postal precej bolj uporaben. Vseeno moramo biti pri uporabi faktorskih modelov nekoliko previdni. Najprej seveda glede besedila, ki ga želimo prevajati, pa tudi glede na vrsto modela, ki ga izberemo. Kot smo pokazali v nalogi, bolj kompleksen model s prerazporejanjem, več koraki prevajanja in ustvarjanja ne pomeni boljših prevodov. Izbrati moramo torej pravi model za svoj namen, kar pa ni vedno preprosto.

Kako bi lahko naše modele izboljšali? Prvi in najbolj očiten način je uporaba večjega korpusa. Statistično strojno prevajanje je odvisno od podatkov, ki jih ima na voljo, zato bi z obsežnejšimi korpusi modeli delovali bolje. Korpus, s katerim smo delali, je imel 2,4 milijona vrstic, kar je za to vrsto prevajanja relativno malo. Prevodi opisov tiskalnikov so bili kljub temu dobri, ker je bilo besedilo podobno tistemu iz korpusa. Če bi želeli prevajati splošna besedila, bi potrebovali bistveno večji korpus. Drugi način bi bil uporaba drugačnega nabora za preskušanje. Naš je imel 200 vrstic, če pa bi uporabili takega z nekaj tisoč vrsticami, bi bili prevodi nekoliko boljši, vendar bi to imelo manjši vpliv kot večji korpus ustreznih prevodov. Že v našem primeru smo za prilagajanje kompleksnih faktorskih modelov potrebovali 10-12 ur na zmogljivem fakultetnem strežniku, v primeru večjega nabora za prilagajanje bi bil ta čas temu ustrezno daljši. Poleg velikosti nabora je pomembna tudi njegova vsebina, ki mora biti čim bolj podobna besedilu, ki ga prevajamo. Naš nabor je bil zelo podoben prevajanemu tehničnemu, manj pa tržnemu besedilu. Če bi za vsako besedilo uporabili drugačen nabor, bi bili rezultati nekoliko boljši, ker pa je bil naš glavni namen primerjati modele med seboj, nam je omenjeni nabor za prilagajanje zadostoval.

Primerjali smo le majhno število modelov, ki jih omogoča Moses. Želeli smo namreč narediti preizkus možnosti in postopno dodajati funkcije ter tako ugotovljati, kako vplivajo na kakovost prevodov. Na ta način smo lahko preskusili le omejen nabor možnosti in parametrov, ki jih omogoča Moses – kombinacij jezikovnih modelov, prerazporejanja, izbranih faktorjev, korakov prevajanja in ustvarjanja ter prevajalnih modelov, ki bi jih lahko preskusili, je namreč veliko. Izkazalo se je, da vsi modeli niso primerni za prevajanje, saj je treba paziti, da so koraki prevajanja in ustvarjanja med seboj neodvisni ter da uporabljajo ustrezne jezikovne modele. Tako smo pri nekaterih modelih dobili neprevedeno izhodno besedilo ali pa je bilo v prevodu nenavadno veliko število ločil, na primer v stavku »*ko je executed with care abstract photography*) , , , , ki jih je mogoče *hugely creative* «.

Videti je tudi, da bi bila dobra rešitev uporabiti hibridni model, kjer bi združili faktorsko statistično strojno prevajanje in strojno prevajanje, ki temelji na pravilih. Če pogledamo prevod stavka »*To be able to really give an honest account of a city you need to fully understand its make up.*«, vidimo, da je preveden kot »*da boste lahko zares zagotavlja pristno račun mesta, kar potrebujete, če želite povsem razumemo, da je njegova navzgor.*«. V dvojezičnem korpusu je tak primer razbit na dva dela, zato se prevod začne z vejico. Če bi imeli v modelu pravilo, ki bi določalo, da je to začetek stavka, takih težav ne bi imeli. Z dodajanjem pravil bi odpravili tudi težave z nizi, v katerih se pojavljajo oklepaji (besedilo v oklepajih v našem primeru ni prevedeno), in s števili, kjer moramo vejice za tisočicami v angleščini zamenjati s pikami v slovenščini (naši modeli so pustili vejice).

Pregled naših prevodov je opravilo 5 jezikoslovcev, saj smo želeli tudi mnenje o slovnični ustreznosti. Vseeno bi lahko pregled dopolnili tako, da bi ga opravile tudi osebe brez jezikoslovnega ozadja, ki jim je strojni prevod pogosteje namenjen, saj velja, da manj jezikovnega znanja, kot imajo uporabniki, bolj uporabno je strojno prevajanje [26].

Literatura

- [1] A. Ratnaparkhi, A maximum entropy model for part-of-speech tagging, Proceedings of the Conference on Empirical Methods in Natural Language Processing, 1996. Dostopno na: <http://www.anthology.aclweb.org/W/W96/W96-0213.pdf>.
- [2] C. Zaiontz, Performing Real Statistical Analysis Using Excel: <http://www.real-statistics.com> [Dostop: julij 2016]
- [3] D. Jurafsky, J.H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Prentice-Hall, 2000.
- [4] European Commission > Automated Translation: <https://ec.europa.eu/digital-single-market/en/automated-translation> [Dostop: avgust 2016].
- [5] F. J. Och, Minimum error rate training in statistical machine translation, Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), str. 160–167, 2003. Dostopno na: <http://www.aclweb.org/anthology/P03-1021>.
- [6] J. R. Landis, G. G. Koch, The Measurement of Observer Agreement for Categorical Data, Biometrics, International Biometric Society, 1977.
- [7] J. Toporišič, Slovenska slovnica, str. 255, Založba Obzorja, 2004
- [8] K. Knight, A Statistical MT Tutorial Workbook, JHU Summer Workshop, Johns Hopkins University, 1999. Dostopno na: <http://www.isi.edu/natural-language/mt/wkbk.rtf>.
- [9] K. Papineni, S. Roukos, T. Ward in W. Zhu, BLEU: a Method for Automatic Evaluation of Machine Translation, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002.
- [10] Kenneth Heafield, kenlm code: <http://kheafield.com/code/kenlm/> [Dostop: maj 2016].
- [11] Moses - FactoredTraining/BuildingLanguageModel: <http://www.statmt.org/moses/?n=FactoredTraining.BuildingLanguageModel> [Dostop: maj 2016].
- [12] Moses – Moses/Background: <http://www.statmt.org/moses/?n=Moses.Background> [Dostop: maj 2016].
- [13] Moses – Moses/FactoredModels: <http://www.statmt.org/moses/?n=Moses.FactoredModels> [Dostop: maj 2016].
- [14] Moses - Moses/Overview: <http://www.statmt.org/moses/?n=Moses.Overview> [Dostop: maj 2016].
- [15] MosesCore Main/Home Page: <http://www.statmt.org/mosescore/> [Dostop: julij 2016].
- [16] OZNAČEVALNIK | Sporazumevanje - OZNAČEVALNIK: <http://www.slovenscina.eu/tehnologije/oznacevalnik> [Dostop: maj 2016].

- [17] P. Koehn, F. J. Och, D. Marcu, Statistical Phrase-Based Translation, Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, str. 127–133, University of Southern California, 2003.
- [18] P. Koehn, Statistical Machine Translation, Cambridge University Press, 2010.
- [19] Penn Treebank P.O.S. Tags:
https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html [Dostop: maj 2016]
- [20] T. Erjavec, S. Krek, Projekt JOS, Zbornik Šeste konference Jezikovne tehnologije, Ljubljana, 2008.
- [21] Software – Digital Silk Road: <http://www.digitalsilkroad.net/software.html> [Dostop: maj 2016].
- [22] M. Romih, S. Krek, I. Kosem, Sporazumevanje v slovenskem jeziku:
<http://www.slovenscina.eu/> [Dostop: maj 2016].
- [23] Support for Moses, an open source statistical machine translation package:
<http://comments.gmane.org/gmane.comp.nlp.moses.user/4255> [Dostop: maj 2016].
- [24] Š. Vintar, Računalniške tehnologije za prevajanje, Uporabna Informatika, Slovensko društvo Informatika, 1999.
- [25] T. Erjavec, S. Krek, Š. Arhar, D. Fišer, N. Ledinek, A. Saksida, B. Sivec, B. Trebar, Oblikoskladenjske specifikacije, Zbornik Šeste konference Jezikovne tehnologije, Ljubljana, 2008.
- [26] W. Hutchins, Uses and Applications of Machine Translation, Presentation at Westminster University, 2009. Dostopno na: www.hutchinsweb.me.uk/Westminster-2009.pdf.
- [27] Warren Weaver, »Translation«, v W. N. Locke in A. D. Booth, Machine Translation of Languages, MIT Press, 1955.
- [28] Wikipedia, »Cohen's kappa«: https://en.wikipedia.org/wiki/Cohen%27s_kappa [Dostop: julij 2016]
- [29] Wikipedia, »Fleiss' kappa«: https://en.wikipedia.org/wiki/Fleiss%27_kappa [Dostop: julij 2016]
- [30] Wikipedia, »IBM alignment models«:
https://en.wikipedia.org/wiki/IBM_alignment_models [Dostop: maj 2016].
- [31] Wikipedia, »Inter-rater reliability«: https://en.wikipedia.org/wiki/Inter-rater_reliability [Dostop: julij 2016]
- [32] Wikipedia, »LEPOR«: <https://en.wikipedia.org/wiki/LEPOR> [Dostop: julij 2016].
- [33] Wikipedia, »Machine translation«: https://en.wikipedia.org/wiki/Machine_translation [Dostop: maj 2016].
- [34] Wikipedia, »METEOR«: <https://en.wikipedia.org/wiki/METEOR> [Dostop: julij 2016].

[35] Wikipedia, »NIST (metric)«: [https://en.wikipedia.org/wiki/NIST_\(metric\)](https://en.wikipedia.org/wiki/NIST_(metric)) [Dostop: julij 2016].

[36] Wikipedija, »Statistično strojno prevajanje«:

https://sl.wikipedia.org/wiki/Statisti%C4%8Dno_strojno_prevajanje [Dostop: maj 2016].

[37] Wikipedija, »Lema«: <https://sl.wikipedia.org/wiki/Lema> [Dostop: maj 2016].