UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

ZAKLJUČNA NALOGA

IZDELAVA STATISTIČNEGA STROJNEGA PREVAJALNIKA IZ ITALIJANŠČINE V SLOVENŠČINO

JANI SUBAN

2022
ZAKLJUČNA NALOGA
SUBAN

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga

Izdelava statističnega strojnega prevajalnika iz italijanščine v slovenščino

(Implementation of statistical machine translation from Italian into Slovenian)

Ime in priimek: Jani Suban
Študijski program: Računalništvo in informatika
Mentor: izr. prof. dr. Jernej Vičič
Somentor: asist. Aleksandar Tošić

Koper, avgust 2022

Ključna dokumentacijska informacija

Ime in PRIIMEK: Jani SUBAN

Naslov zaključne naloge: Izdelava statističnega strojnega prevajalnika iz italijanščine v slovenščino

Kraj: Koper

Leto: 2022

Število listov: 64 Število slik: 12 Število tabel: 15

Število prilog: 1 Število strani prilog: 4 Število referenc: 26

Mentor: izr. prof. dr. Jernej Vičič

Somentor: asist. Aleksandar Tošić

Ključne besede: Strojno prevajanje, statistično strojno prevajanje, SMT, prevajanje

Izvleček:
V zaključni nalogi bo predstavljena implementacija statističnega strojnega prevajanja. Na začetku bom opisal strojno prevajanje in njegove obstoječe metode. Naloga bo osredotočena na statistično strojno prevajanje in njegovo delovanje. Predstavljeni bodo tudi že obstoječi prevajalniki. Zatem bo predstavljena implementacija statističnega strojnega prevajanja iz italijanščine v slovenščino. Implementirana bo s pomočjo odprtokodnega sistema Moses, ki je nastal v sklopu projekta EuroMatrixPlus. Nato bom primerjal prej opisano implementacijo s prevajalniki, ki so bili že predhodno predstavljeni v zaključni nalogi.

Key document information

Name and SURNAME: Jani SUBAN

Title of the final project paper: Implementation of statistical machine translation from Italian into Slovenian

Place: Koper

Year: 2022

Number of pages: 64 Number of figures: 12 Number of tables: 15

Number of appendices: 1 Number of appendix pages: 4 Number of references: 26

Mentor: Assoc. Prof. Jernej Vičič, PhD

Co-Mentor: Assist. Aleksandar Tošić

Keywords: Machine translations, statistical machine translations, SMT, translation

Abstract:
The thesis will present the implementation of statistical machine translation. First, it explains what machine translation is and what methods of machine translation exist. The focus of the thesis will be on statistical machine translation and how it works is described. In addition, the existing translators are introduced. This is followed by the implementation of statistical machine translation from Italian into Slovenian. It is implemented using the open-source system Moses, which was developed as part of the EuroMatrixPlus project. Afterwards, I will compare the previously described implementation with the translators described earlier in this thesis.

Zahvala

Najlepše bi se rad zahvalil mentorju, izr. prof. dr. Jerneju Vičiču, in somentorju, asist. Aleksandru Tošiću, za pomoč pri izdelavi zaključne naloge. Poleg tega bi se rad zahvalil tudi svoji družini in prijateljem. Posebno pa bi se rad zahvalil svojim staršem in bratu, ki so mi bili v pomoč ter mi stali ob strani v času študija.

Kazalo vsebine

Kazalo tabel

Kazalo slik in grafikonov

Kazalo prilog

Seznam kratic

MT	Ang. Machine Translation (slo. Strojno prevajanje)
SMT	Ang. Statistical Machine Translation (slo. Statistično strojno prevajanje)
PBMT	Ang. Phrase-based Machine Translation (slo. Strojno prevajanje s pomočjo fraz)
NMT	Ang. Neural Machine Translation (slo. Strojno prevajanje na osnovi nevronskih mrež)
RBMT	Ang. Rule-based Machine Translation (slo. Strojno prevajanje na osnovi pravil)
EBMT	Ang. Example-based Machine Translation (slo. Strojno prevajanje na osnovi primerov)
HMT	Ang. Hybrid Machine Translation (slo. Hibridno strojno prevajanje)
RAM	Ang. Random Access Memory (slo. Bralno-pisalni pomnilnik)
EU	Evropska unija
OOV	Ang. Out-of-vocabular (slo. Neznana beseda)
RNN	Ang. Recurrent Neural Network (slo. Nevronska mreža s povratno povezavo)
BLEU	Ang. Bilingual Evaluation Understudy
DARPA	Ang. Defense Advanced Research Projects Agency (slo. Agencija za napredne obrambne analize)
WBMT	Ang. Word-based Machine Translation (slo. Strojno prevajanje na osnovi besed)
PBMT	Ang. Phrase-based Machine Translation (slo. Strojno prevajanje na osnovi fraz)
GNMT	Ang. Google Neural Machine Translation
XML	Ang. Extensible Markup Language
TMX	Ang. Translation Memory eXchange

MERT	Ang. Minimum error rate training
CMPH	Ang. C Minimal Perfect Hashin
API	Ang. Application Programming Interface (slo. Aplikacijski programski vmesnik)

1 Uvod

Prevajanje je proces, pri katerem besedilo, bodisi pisno bodisi govorjeno, spremenimo iz izhodnega v ciljni jezik. Pri tem se ohrani pomen besedila v obeh jezikih. Ideja o prevajanju ni nova, saj so že stari Egipčani prevajali svoja besedila. Primer takih prevodov je Kamen iz Rosette [20], kjer je zapisano isto besedilo v treh različnih jezikih. Strojno prevajanje (Ang. Machine Translation, MT) je način prevajanja, kjer ni neposrednega človeškega stika pri izdelavi prevoda. Začelo se je razvijati v drugi polovici 20. stoletja z razvojem interneta ter povečanjem računske sposobnosti računalnikov.

Cilj zaključne naloge je izdelava prevajalnika, ki bo prevedel besedilo iz italijanskega jezika v slovenski. Prevajalnik bo izdelan z metodo statističnega strojnega prevajanja (Ang. Statistical Machine Translation, SMT) [19]. Jezika sta bila izbrana zato, ker ne sodita v isto jezikovno družino: slovenščina spada med slovanske jezike, italijanščina pa med romanske jezike. Zato si nista podobna. Poleg tega ta jezikovni par ni pogosto uporabljen pri strojnem prevajanju, kot na so primer: angleščina in francoščina, italijanščina in angleščina ali angleščina in slovenščina. Na Sliki 1 je prikazan bločni diagram prevajalnika, kjer i predstavlja izvorno poved v italijanščini, s pa ciljno poved v slovenščini.

Slika 1: Splošni diagram prevajalnika

Zaključna naloga je razdeljena na tri poglavja. V prvem poglavju so predstavljeni in definirani strojno prevajanje ter njegove metode prevajanja. V nadaljevanju zaključne naloge je podrobno opisano statistično strojno prevajanje, prav tako pa je tudi podrobno predstavljena matematična podlaga za tako prevajanje. Zatem so predstavljeni že obstoječi prevajalniki, bolj podrobno sta opisana prevajalnika DeepL [13] in Google Prevajalnik [14, 15].

V naslednjem poglavju so predstavljeni implementacija statističnega strojnega prevajanja, sistem za statistično strojno prevajanje Moses ter korpusi, ki so bili uporabljeni za izdelavo prevajalnika.

V zadnjem poglavju pa so zajete metode in rezultati evalvacija sistema. Predstavljena je metrika BLEU (Ang. Bilingual Evaluation Understudy) [5] za avtomatizirano ocenjevanje prevajalnikov. Poleg tega so podani rezultati primerjave med različnimi prevajalniki in človeškim prevodom.

2 Statistično strojno prevajanje

Poglavje je osredotočeno na splošno predstavitev statističnega strojnega prevajanja in je razdeljeno na tri sklope.

V prvem sklopu je zajet opis strojnega prevajanja, kjer so naštete in na kratko opisane različne metode strojnega prevajanja. Prav tako sta podrobno predstavljena statistično stojno prevajanje ter njegov razvoj.

Drugi sklop poglavja vsebuje matematična orodja, na katerih temelji statistično strojno prevajanje. Predstavljena sta uporaba Bayesovega izreka v strojnem prevajanju in postopek izbire najboljšega prevoda, ki je uporabljen v sistemu Moses.

V tretjem sklopu poglavja pa je podan opis že obstoječih prevajalnikov, in sicer bosta zelo natančno sistema Google Prevajalnik in DeepL. Sistema bosta tudi uporabljena za primerjavo s sistemom, predstavljenim v poglavju 3 Implementacija.

2.1 Strojno prevajanje

Strojno prevajanje je način prevajanja, pri katerem človek ni neposredno prisoten. Zato se ne uvršča med programe za računalniško podprto prevajanje, med katere spadajo: elektronski slovarji, pregledovalniki slovnice, konkordančniki in tako dalje. Strojno prevajanje se lahko uvrsti v to kategorijo le, ko človek odpravi nejasnosti v prevodu.

Zametki strojnega prevajanja izhajajo iz druge polovice 20. stoletja. Ena prvih omemb strojnega prevajanja je iz leta 1949 in je zajeta v pismu Warrena Weaverja. To je le štiri leta po izgradnji prvega ENIAC-a. Od takrat je strojno prevajanje eden glavnih problemov, ki se raziskuje v sklopu jezikovnih tehnologij.

V tem času se je razvilo več metod strojnega prevajanja, in sicer:

- Strojno prevajanje na osnovi pravil (Ang. Rule-based Machine Translation, RBMT), kjer se besedilo razdeli na besede, ki se bodo prevedle posamično. Pravila za take prevajalnike so dvojezični slovarji za izbran jezikovni par.

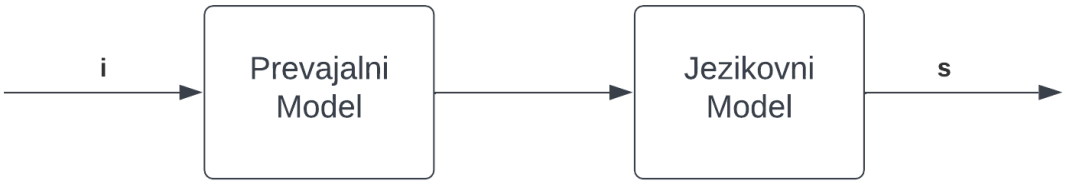
- Strojno prevajanje na osnovi primerov (Ang. Example-based Machine Translation, EBMT), kjer se besedilo razdeli na besedne zveze, ki se nato iščejo v korpusu in se z njegovo pomočjo tudi prevedejo.

- Statistično strojno prevajanje (Ang. Statistical Machine Translation, SMT) [19].

- Strojno prevajanje na osnovi nevronskih mrež (Ang. Neural Machine Translation, NMT) [26], je trenutno najbolj razširjena metoda, saj izdela primerljive oziroma boljše prevode kot SMT, pri čemer potrebuje manj spomina za delovanje. Za izdelavo prevodov NMT uporablja nevronske mreže s povratno povezavo (Ang. Recurrent Neural Network, RNN) [14].

- Hibridno strojno prevajanje (Ang. Hybrid Machine Translation, HMT) uporablja za izboljšavo prevodov dva (ali več) različna pristopa strojnega prevajanja. Najbolj pogosto uporabljen par sta SMT in RBMT.

V nadaljevanju besedila bo podrobno opisano statistično strojno prevajanje, ki je tema te zaključne naloge. Temeljna ideja tega pristopa je analiza parov besedil in iskanje vzorcev v njih. To je storjeno s pomočjo Bayesovega izreka. Na Sliki 2 je prikazan bločni diagram tega pristopa, pri čemer i predstavlja izvorno poved v italijanščini, s pa ciljno poved v slovenščini. Povezava med primerom, prikazanim na Sliki 2, in Bayesovim izrekom [4] bo predstavljena v naslednjem poglavju, 2.2 Matematična Podlaga.



Slika 2: Splošni diagram prevajalnika, ki uporablja SMT

Razvoj statističnega strojnega prevajanja se je začel v 80. letih prejšnjega stoletja. V naslednjih 30. letih je postal prevladujoča različica strojnega prevajanja, tako na akademskem področju kot tudi v praktični implementaciji. To se je zgodilo, ker je implementacija prevajalnika SMT zelo hitra, prevodi pa dosegajo visoko kakovost. Znižanje časa implementacije je prispeval tudi razvoj orodij (Ang. Toolkits), kot je na primer Moses [6, 11]. Visoka kakovost prevodov pa je bila dosežena na račun uporabe velike količine dvojezičnih korpusov oziroma besedil. Te morajo predstavljati kakovostni prevodi, ki so posebej izbrani za namen prevajalnika. Danes ga je nadomestilo strojno prevajanje na osnovi nevronskih mrež (NMT) tako na akademskem področju kot tudi v praktični implementaciji, saj ta metoda potrebuje manjše število korpusov.

Razlogi za razširjenost SMT so državne pomoči za razvoj same tehnologije, razvoj interneta ter padec cen računalniških delov. Predvsem padec cen delovnih pomnilnikov (Ang. Random Access Memory, RAM) je omogočil uporabo večjih korpusov pri

izdelavi prevajalnikov. Razvoj interneta je omogočil ljudem dostop do informacij, ki niso zapisane zgolj v njihovem matičnem jeziku. Tako se je začela razvijati ideja o prevajalnikih, ki bi prevajali vsebino v matični jezik uporabnikov interneta. Poleg tega so tudi različne države oziroma skupine držav, kot so na primer Evropska unija in Združene države Amerike, začele razvijati ali sofinancirati razvoj prevajalnikov. EU je s tem namenom razpisala projekta EuroMatrix, ki je potekal od septembra 2006 do februarja 2009, in EuroMatrixPlus, ki pa je potekal od marca 2009 do februarja 2012. Namen teh dveh projektov je bila izboljšava stanja strojnega prevajanja na evropskem področju. To je bilo storjeno z izdelavo korpusov v vseh evropskih jezikih ter z izdelavo in izboljšavo orodji za strojno prevajanje, kot sta na primer Moses in IRSTLM. V ZDA je podoben projekt razpisala Agencija za napredne obrambne analize (Ang. Defense Advanced Research Projects Agency, DARPA), ki se je osredotočila na razvoju SMT. Kljub temu se je ta tehnologija v zadnjem desetletju začela opuščati, saj se je NMT izkazal za boljši pristop k strojnemu prevajanju.

2.2 Matematična podlaga

Diagram prikazan na Sliki 1, se lahko zapiše z uporabo enačbe:

$$s = Prev(i), \tag{2.1}$$

pri čemer je i poved v italijanskem jeziku, s pa predstavila poved v slovenskem jeziku.[1] Za poved i velja, da je $i = i_0 i_1 i_2 \dots i_l;\ i_j \in I,\ j \in \{0, 1, \dots, l\}$, pri čemer je l dolžina povedi, I pa so vse besede v italijanščini. Podobno velja za s, kjer je $s = s_0 s_1 s_2 \dots s_k;\ s_j \in S,\ j \in \{0, 1, \dots, k\}$, pri čemer je k dolžina povedi, S pa so vse besede v slovenščini. V nadaljevanju naloge bodo vse vrednosti i in s imele zgoraj navedene lastnosti [10].

Kot je bilo že omenjeno v poglavju 2.1 Strojno prevajanje, SMT išče vzorce v besedilu, jim priredi verjetnost ter izdela prevod, ki vsebuje najbolj verjetno slovensko poved. Pri tem SMT uporablja Bayesov izrek:

$$P(s|i) = \frac{P(i|s)P(s)}{P(i)}. \tag{2.2}$$

Pri tem je $P(i) = 1$, saj se predpostavi, da je poved v italijanščini vedno pravilna. $P(s)$ se imenuje statistični jezikovni model, $P(i|s)$ pa se imenuje statistični prevajalni model. Ta dva modela sta prikazana tudi na Sliki 2. Statistični jezikovni model predstavlja verjetnost obstoja povedi v slovenskem jeziku. Statistični prevajalni model pa

[1] Nomenklatura povedi v italijanskem in slovenskemu jeziku, namesto bolj splošne povedi v izvirnem in ciljnem jeziku, je bila izbrana, saj naloga se osredotoči le na ta jezikovni par.

predstavlja verjetnost prevoda povedi i v poved s. Torej, če se enačbo (2.2) preveri za vse možne s in se izbere tisti s z največjo vrednostjo $P(s|i)$, se dobi enačbo

$$\widehat{s} = \underset{s}{\operatorname{argmax}} P(i|s)P(s), \tag{2.3}$$

pri čemer je $\widehat{s}$, najboljša poved v slovenščini, ki ima isti pomen kot poved i [9]. Za umerjanje dolžine prevoda (števila besed v povedi) se enačbo (2.3) spremeni v

$$\widehat{s} = \underset{s}{\operatorname{argmax}} P(i|s)P(s)\omega^{|s|}, \tag{2.4}$$

kjer se uvede faktor ω, ki se mu reče cena besede. Na ta način izboljšamo delovanje modela. Po navadi je $\omega > 1$, kar daje prednost daljšim izhodnim povedim [3, 6].

V nadaljevanju bosta še podrobnejše opisana statistični jezikovni model ter statistični prevajalni model. Predstavljena modela bosta uporabljena pri implementaciji SMT-ja v poglavju 3 Implementacija. Poleg tega bo tudi predstavljeno delovanje faze poravnave besedil in faze dekoder.

2.2.1 Jezikovni model

Namen statističnega jezikovnega modela je izdelava čim bolj slovnično pravilnih povedi v slovenščini (ciljni jezik prevajanja). Tako se izogne povedim, katerih posamične besede oziroma besedne zveze ohranijo pomen pri prevodu iz italijanščine v slovenščino, vendar celotna prevedena poved nima pomena v slovenščini.

Statistični jezikovni model je definiran kot $P(s)$, kar predstavlja verjetnost obstoja povedi s v slovenščini, in se ga lahko zapiše kot:

$$P(s) = \prod_{j=1}^{k} P(s_j|s_0, s_1, ..., s_{j-1}) \tag{2.5}$$

kjer je uporabljeno verižno pravilo za pogojno verjetnost [9]. Statistični jezikovni model, ki je uporabljen za izdelavo sistema, uporablja n-terke. Zato se lahko enačbo (2.5) zapiše kot:

$$P(s) \approx \prod_{j=1}^{k} P(s_j|s_{j-(n-1)}, ..., s_{j-1}). \tag{2.6}$$

V pristopu, ki uporablja n-terke, vpliva na izbiro besede zgolj predhodnih $n-1$ besed v povedi s, za razliko od splošnega pristopa, kjer vplivajo vse predhodne besede v povedi s [10]. Na ta način se poenostavi statistični jezikovni model, pri čemer pa se ne zmanjša kakovosti izbire pravilnega prevoda.

2.2.2 Prevajalni model

Za razliko od statističnega jezikovnega modela je namen statističnega prevajalnega modela izdelava čim bolj vsebinsko/pomensko natančnih prevodov. To se lahko doseže na dva načina: prvi način je strojno prevajanje na osnovi besed (Ang. Word-based Machine Translation ali WBMT), drugi način pa je strojno prevajanje na osnovi fraz (Ang. Phrase-based Machine Translation ali PBMT) [3], pri čemer fraza predstavlja zaporedje besed. Pristop PBMT se lahko pretvori v pristop WBMT, tako da se pri prevajanju se upošteva zgolj fraze dolžine 1, kar so besede. Predstavljen bo zgolj pristop PBMT, saj bo uporabljen v nadaljevanju zaključne naloge.

Pri PBMT se najprej razdeli poved i na J fraz $\bar{i}_1^J$, pri čemer se predpostavi, da so vse možne fraze enakomerno porazdeljene. Statistični prevajalni model je definiran kot verjetnost $P(i|s)$, torej se ga lahko zapiše kot:

$$P\left(\bar{i}_1^J|\bar{s}_1^J\right) = \prod_{j=1}^{J} \Phi\left(\bar{i}_j|\bar{s}_j\right) d\left(a_j - b_{j-1}\right), \tag{2.7}$$

kjer je $\Phi\left(\bar{i}_j|\bar{s}_j\right)$ porazdelitvena funkcija verjetnosti prevoda fraze $\bar{i}_j$ v frazo $\bar{s}_j$. Izračunana je kot relativna verjetnost [3]:

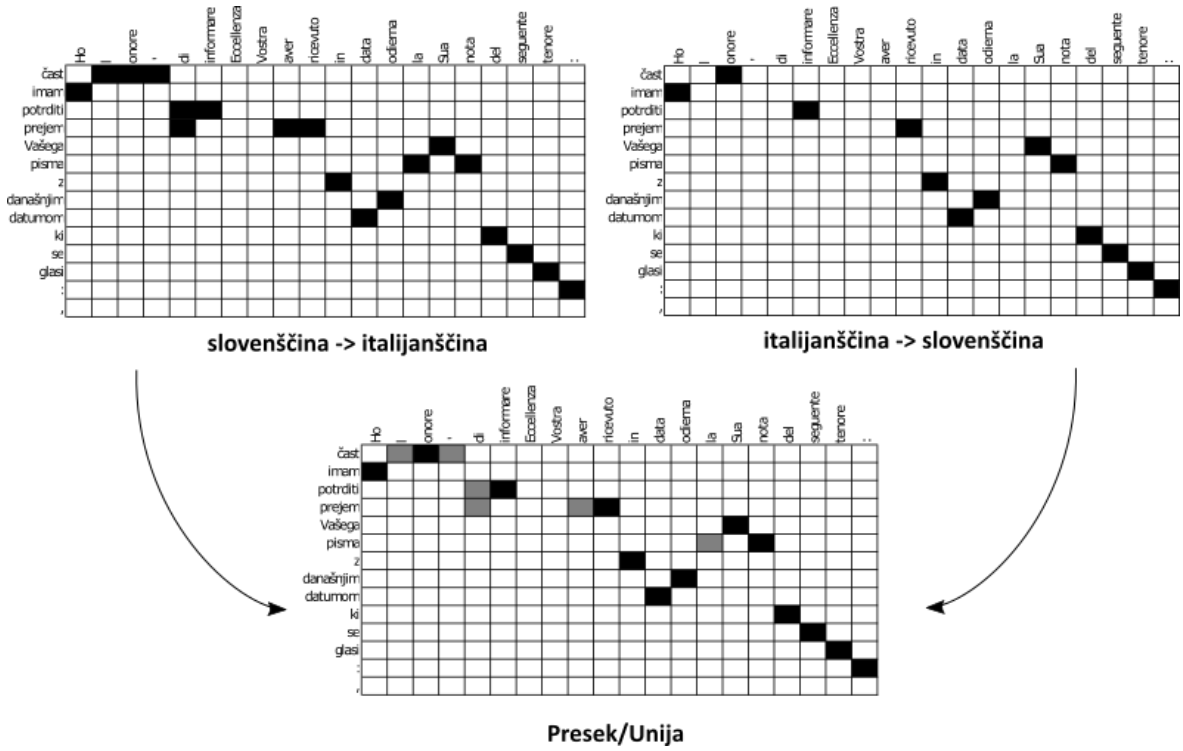
$$\Phi\left(\bar{i}_j|\bar{s}_j\right) = \frac{count(\bar{i}|\bar{s})}{\sum_{\bar{i}} count(\bar{i}|\bar{s})}. \tag{2.8}$$

Pri tem se lahko frazo v slovenščini še naknadno preoblikuje. Preoblikovanje slovenskih fraz je modelirano s pomočjo funkcije izkrivljenja (Ang. distortion function). S funkcijo izkrivljenja $d(a_j - b_{j-1})$, v kateri a_j predstavlja začetek fraze v slovenščini, ki se prevede iz j-te fraze v italijanščini, in b_{j-1} konec fraze v slovenščini, ki se prevede iz $(j-1)$-te fraze v italijanščini. Funkcija izkrivljenja, ki je uporabljana v sistemu Moses, je definirana kot $d(a_j - b_{j-1}) = \alpha^{|a_j - b_{j-1} - 1|}$. Pri tem je izbrana primerna vrednost za parameter α [6].

2.2.3 Poravnava besed

Poravnava besed (Ang. Word Alignment) je začetni korak statističnega strojnega prevajanja, saj se s poravnavo besed izračuna verjetnosti, ki so potrebne za izdelavo statističnega prevajalnega modela. Orodje, ki je uporabljeno v nalogi za poravnavo besed, je GIZA++ [8], ki je trenutno tudi najbolj uporabljeno orodje in je implementacija IBM-ovih modelov. Ti modeli so [8]:

- IBM-1: je najbolj preprost model in predpostavi, da je vsaka poravnava enako verjetna.
- IBM-2: je nadgradnja IBM-1, pri čemer omogoča preoblikovanje povedi. Pri tem imajo različne povedi različno verjetnost ter so med seboj neodvisne.
- IBM-3: je nadgradnja IBM-2, ki omogoča dodajanje besed pri poravnavi besedila. Primer takega prevoda je prevod italijanske besede »andavo«, ki se prevede v slovenščino kot »sem šel«.
- IBM-4: je nadgradnja IBM-3, pri čemer upošteva medsebojno odvisnost besed, kot je na primer odvisnost med pridevniki in samostalniki. Pri tem se upošteva zgolj odvisnost do predhodne besede.
- IBM-5: je reformulacija IBM-4 z izboljšano poravnavo besede. Implementacija omogoča dodajanje še neporavnanih besed na mesta, ki niso še zasedena.



Slika 3: Primer poravnave besed

Poravnava besed se začne z obojestransko poravnavo dvojezičnega korpusa. Tako sta dobljeni dve različni poravnavi, in sicer iz slovenščine v italijanščino ter iz italijanščine v slovenščino. To je razvidno iz zgornjega dela Slike 3, ki prikazuje primer poravnave povedi »čast imam potrditi prejem Vašega pisma z današnjim datumom, ki se glasi:« (Ita. »Ho l'onore di informare l'Eccellenza Vostra di aver ricevuto in data odierna la Sua nota del seguente tenore:«), ki je 115. vpis v korpusu JRC Acquis.

Naslednji korak je združenje obeh poravnav. To je storjeno na dva načina. Prvi način je s presekom obeh poravnav, ki je prikazan s črno barvo na spodnjem delu Slike 3. Presek prikaže poravnave z visoko natančnostjo (Ang. precision). Drugi način pa temelji na uniji obeh poravnav, ki je prikazana s sivo in črno barvo na spodnjem delu Slike 3. Unija pa nam prikaže poravnavo z visokim priklicem (Ang. recall) [6].

Večina metod za poravnavo besed uporablja zgoraj opisani postopek. Postopek, ki bo predstavljen v nadaljevanju, je implementacija, ki je uporabljena v sistemu Moses. Ta implementacija temelji na implementaciji, predstavljeni v članku [8].

Metoda je prikazana s psevdokodo, zapisano v Algoritmu 1. Algoritem sprejme kot vhodne podatke presek obeh poravnav, ki je v algoritmu zapisan kot A_P, ter unijo obeh poravnav, ki pa je v algoritmu zapisana kot A_U. Predstavljen algoritem pa vrne končno poravnavo, ki je zabeležena kot A. Končna poravnava vedno vsebuje presek. Ideja implementirane metode je sprotno dodajanje novih poravnav iz A_U v A.

Algoritem 1: Delovanje poravnave besed v sistemu Moses

Vhod: A_P - Presek obeh poravnav
A_U - Unija obeh poravnav
Izhod: A - Končna poravnava

```
1  A = A_P;
2  delaj
3      x = False;
4      za e ∈ A_U \ A
5          če e je oddaljen eno mesto od elementoma v A, potem
6              A = A ∪ {e};
7              x = True;
8  dokler x;
9  za e ∈ A_U \ A
10     če e_1 ALI e_2 ni poravnan, potem
11         A = A ∪ {e};
12         x = True;
13 vrni A;
```

V vsaki iteraciji algoritma se poravnajo slovenske besede, ki še niso poravnane. Pri tem se vedno začne pri prvi neporavnani slovenski besedi. Iskanje poravnav besede se opravi pri sosedih, pri tem se začne pri zgornjem desnem sosedu. Najprej se dodajo vse poravnave iz A_U, ki se stikajo s poravnavami iz A (so oddaljeni le eno mesto v tabeli). Postopek je prikazan med 2. in 8. vrstico v Algoritmu 1. Ta korak se ponavlja, dokler ni mogoče dodati nove poravnave iz $A_U \setminus A$ v A, kar se preverja z zastavico x. V naslednjem koraku se dodajo še vse poravnave, kjer vsaj ena beseda ni poravna (bodisi

e_1 bodisi e_2). Postopek je predstavljen med 9. in 12. vrstico v Algoritmu 1. Pri tem veljajo isti pogoji kot v prejšnjem koraku.

Iz poravnave A se lahko sestavi množico vseh dvojezičnih fraz ali **DF**, s pomočjo katere se bo lahko izračunalo porazdelitev fraz iz enačbe (2.8). Množica vseh dvojezičnih fraz je definirana kot [6]:

$$\mathbf{DF}(\bar{i}_1^J, \bar{s}_1^J, A) = \{(i_j^{j+m}, s_k^{k+n}); \forall (k', j') \in A; j \leq j' \leq j+m \Leftrightarrow k \leq k' \leq k+n\}, \quad (2.9)$$

pri čemer m predstavlja besedo v italijanščini, n pa besedo v slovenščini. Na ta način se lahko poved, ki je bila uporabljena kot primer na Sliki 3, razdeli v fraze. Prvih 9 fraz ter zadnja fraza te povedi so predstavljene v Tabeli 1. Na vrhu Tabele 1 so vse fraze, ki so tvorjene iz ene slovenske besede, nato z dvema in tako dalje. Zadnja fraza je vedno celotno izhodiščno besedilo (v tem primeru je poved).

Tabela 1: Dvojezične fraze

Slovenske fraze	Italijanske fraze	Slovenske fraze	Italijanske fraze
čast	l'onore	pisma	la nota
imam	Ho	z	in
potrditi	di informare	današnjim	odierna
prejem	di aver ricevuto	datumom	data
Vašega	Sua	...	...
Slovenske fraze		Italijanske fraze	
čast imam potrditi prejem Vašega pisma z današnjim datumom, ki se glasi:		Ho l'onore di informare l'Eccellenza Vostra di aver ricevuto in data odierna la Sua nota del seguente tenore:	

2.2.4 Dekoder

Po kratki predstavitvi delovanja statističnega prevajalnega modela, statističnega jezikovnega modela ter poravnave besed se lahko prikazane korake združi v postopek, ki je enakovreden enačbi (2.4). V nadaljevanju bo opisan postopek, ki je implementiran v sistemu Moses [6]. Postopek je razdeljen na dva dela, in sicer na izbiro fraz ter na izbiro najboljšega prevoda.

Izbira fraz je postopek izbire vseh možnih fraz za dano vhodno poved, ki se izvrši pred samim prevajanjem. To je treba storiti, saj pri prevajanju ni treba uporabiti vseh fraz, ki jih sistem pozna, ampak zgolj tiste, ki omogočajo prevod vhodne povedi. Na ta načini se pospeši iskanje pravilnega prevoda, saj se v spomin (RAM) ne naloži celotna tabela fraz (vse fraz v sistemu), ker je lahko prevelika, da bi jo v celoti lahko naložili v

delovni spomin. Pri tem vsaka izbrana fraza hrani podatke o začetni italijanski besedi, končni italijanski besedi, enakovredni slovenski frazi in verjetnosti prevoda fraze.

Izbira najboljšega prevoda je postopek izbire prevoda z največjo verjetnostjo. Postopek vedno izdeluje slovensko poved z leve proti desni. Postopek izdela drevo, kjer vsako vozlišče predstavlja možen prevod, ki je imenovan hipoteza. Prva hipoteza, koren drevesa, vedno ne vsebuje fraze, njena verjetnost pa je 1.

Pri tem mora vsaka hipoteza vsebovati naslednje lastnosti:

- povezavo na predhodno hipotezo,
- že prevedene italijanske besede,
- zadnji dve slovenski besedi, ki sta bili izdelani,
- konec zadnje italijanske fraze,
- zadnjo slovensko frazo,
- verjetnost celotne poti,
- predvideno verjetnost nadaljnje poti.

Predvidena verjetnost nadaljnje poti omogoča lažje nadaljevanje iskanja najboljšega prevoda. Na to vrednost vplivajo tri faktorji: statistični jezikovni model, ki uporablja zadnji dve izdelani slovenski besedi za napovedovanje, statistični prevajalni model ter izkrivljenje (Ang. distortion), ki je izračunano s pomočjo lastnosti (parameter) – konec zadnje italijanske fraze. Iskanje najboljšega prevoda uporablja algoritem za iskanje na grafih – Iskanje v snopu (Ang. Beam search). Končni prevod je izbran med listi drevesa, in sicer list z največjo verjetnostjo celotne poti.

2.3 Drugi Prevajalniki

Z razvojem interneta ter digitalizacijo različnih orodij za prevajanje se je ideja o strojnem prevajanju začela konkretizirati. Zato je veliko podjetij, ki se ukvarjajo z razvojem programske opreme, začelo razvijati sisteme ter orodja za strojno prevajanje. Primer takih podjetij so: Microsoft s svojim prevajalnikom Microsoft Translator (poznan tudi kot Bing Translator) [21], IBM s svojim programom IBM Watson [22], ki ni načeloma znan kot sistem za strojno prevajanje, Google s svojim prevajalnikom Google Prevajalnik [14, 15], Amazon s svojim prevajalnikom Amazon Prevajalnik [23], ter druga podjetja. V zadnjih letih so se začela razvijati podjetja, ki se ukvarjajo zgolj s strojnim prevajanjem. Primer takega podjetja je podjetje DeepL [13].

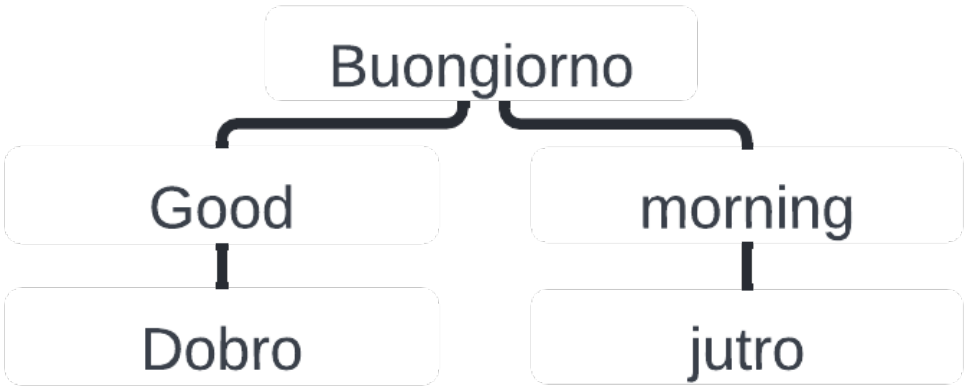
Zaradi majhnega števila slovensko govorečega prebivalstva se zgoraj navedena podjetja sprva niso odloča za izdelavo prevajalnikov, ki prevajajo v ali iz slovenščine. Zato so slovenska podjetja začela samostojno razvijati prevajalnike, ki prevajajo iz ter v slovenščino. Najbolj znan primer takega podjetja je Amebis s svojim prevajalnikom Presis [24]. Poleg tega obstaja slovenski prevajalnik Prevajalnik.si [25], ki je manj znan.

V nadaljevanju bosta bolj podrobno predstavljena prevajalnika Google Prevajalnik ter DeepL. Prevajalnika bosta primerjana v poglavju 4 Evalvacija s predstavljeno implementacijo. Prevajalnika sta bila izbrana, saj uporabljata lasten model za prevajanje. Oba prevajalna sistema podpirata prevajanje med slovenščino in italijanščino za razliko od sistema Presis, ki prevaja iz nemščine, francoščine in angleščine v slovenščino ter iz nemščine, slovenščine in albanščine v angleščino. Googlov Prevajalnik je bil izbran med vsemi omenjenimi prevajalniki zato, ker je najbolj poznan širši javnosti. DeepL pa je bil izbran zaradi trditve, da trenutno ponuja najbolj natančno in kakovostno strojno prevajanje, ki je vsaj 3-krat bolj natančna od konkurence [13].

2.3.1 Google Prevajalnik

Google je prvič predstavil svojo implementacijo strojnega prevajanja 28. aprila 2006 kot orodje, imenovano Google Prevajalnik. Prva implementacija prevajalnika temelji na statističnem strojnem prevajanju (SMT) oziroma bolj natančno uporablja strojno prevajanje na osnovi fraz (PBMT), ki je tudi metoda, uporabljena v sistemu Moses. To implementacijo so uporabljali naslednjih 10 let. Zaradi napredka v razvoju nevronskih mrež se je Google odločil 15. novembra 2016 za prehod na strojno prevajanje na osnovi nevronskih mrež (NMT), imenovano tudi Google Neural Machine Translation (GNMT) [14,15]. Pri tem je bilo uporabljeno odprtokodno Googlovo orodje TensorFlow, ki je eno bolj znanih orodji za implementacijo nevronskih mrež. Sprva je bil NMT implementiran za prevajanje iz španščine, francoščine ter kitajščine v angleščino in iz angleščine v španščino, francoščino ter kitajščino. Do nejasnosti o delovanju Google Prevajalnika

pride zaradi zaprtosti sistema ter zaradi pomanjkanja informacij o delovanju sistema, ki jih Google deli javnosti. V tem času je Google Prevajalnik postal znan, uporabljen in priljubljen med širšo javnostjo.



Slika 4: Primer posrednega prevajanja

S testiranjem Google Prevajalnika je bilo ugotovljeno, da prevajalnik za manj pogoste jezikovne pare ne prevaja neposredno iz izvornega jezika v ciljni jezik, ampak iz izvornega v angleški jezik ter iz angleškega v ciljni jezik. To je potrjeno z dejstvom, da so prevodi vsebovali angleške besede, čeprav izvorni in ciljni jezik nista bila angleščina [2]. Angleške besede so trenutno v prevodih redkeje opažene, kar pomeni, da se je kakovost prevodov izboljšala. Dandanes Google Prevajalnik še vedno ne prevaja neposredno iz izvornega jezika v ciljni jezik. To je prikazano na Sliki 4, in sicer v primeru prevoda italijanske besede »Buongiorno« z Google Prevajalnikom, ki bi se morala prevesti v slovenščino kot »Dober dan«, a se prevede v »Dobro jutro«, saj v angleščini ne obstaja pozdrav »Good day«, ampak obstajata pozdrava »Good morning« in »Good afternoon«.

2.3.2 DeepL

DeepL je nemško podjetje, ki se izključno ukvarja s strojnim prevajanjem. Razlog za izdelavo prevajalnika je želja po kakovostnem prevajanju med evropskimi jeziki. Prevajalnik je bil prvič predstavljen javnosti 28. avgusta 2017 in je prevajal med angleščino, nemščino, francoščino, španščino, poljščino in nizozemščino. Prevajalnik temelji na strojnem prevajanju na osnovi nevronskih mrež (NMT) podobno kot Google Prevajalnik. Dodaten zavoj prevajalnika je 16. marca 2021 doprinesel dodajo 13 novih jezikov, med katerimi je tudi slovenščina. Trenutno DeepL podpira 24 jezikov, pri čemer razlikuje med različicami kitajščine, portugalščine in angleščine, ki pa niso štete posebej, ter 552 jezikovnih parov [13].

V zadnjem obdobju je DeepL pridobil na popularnosti zaradi natančnosti in kakovosti prevodov. Podjetje trdi, da so njihovi prevodi vsaj 3-krat boljši od konkurence (Google Prevajalnik), pri čemer so prevodi med angleščino in nemščino ter japonščino in angleščino kar 6-kart boljši od konkurence ter med angleščino in kitajščino 5-krat boljši od konkurence. Tako kakovost prevodov dosežejo s specifičnostjo učnih podatkov, saj za razliko od Googla, ki uporablja veliko količino učnih podatkov za izdelavo modela, DeepL uporablja manj učnih podatkov, ki so bolj ciljni. Na tak način izboljšajo natančnost prevodov. Ko primer s Slike 4 prevedemo z DeepL, še vedno prevede »Buongiorno« v »Dobro jutro,« a za razliko od Google Prevajalnika ponudi alternativni prevod »Dober dan.«

Osnovna različica DeepL je brezplačna, kot je tudi Google Prevajalnik, ki pa ima nekatere omejitve. Oba prevajalnika omogočata prevajanje besedil do 5000 znakov ter prevajanje dokumentov. Razlikujeta se v tem, da DeepL prevaja (v osnovni različici) zgolj 3 dokumente na mesec ter natančno 5000 znakov naenkrat, Google Prevajalnik pa nima omejitve pri količini prevedenih dokumentov ter besedila, ki presegajo 5000 znakov, razdeli na več delov po 5000 znakov in nato vsak del posebej prevede. Poleg tega Google Prevajalnik omogoča tudi neposredno prevajanje vsebine na spletnih straneh.

3 Implementacija

V tem poglavju sta predstavljeni dve vsebini. Prva predstavljena vsebina je opis sistema Moses. Pri tem bodo predstavljeni tudi vsa uporabljena zunanja orodja ter knjižnice, ki so potrebne za pravilno delovanje sistema.

Drugi del poglavja je osredotočen na izdelavo prevajalnika. Podrobno bodo opisane posamične točke, ki so potrebne potrebne za izdelavo prevajalnika. Predstavljeni bodo tudi vsi korpusi, ki so bili uporabljeni pri izdelavi prevajalnika.

3.1 Moses

Moses je odprtokodna implementacija statističnega strojnega prevajanja (SMT). Sistem se je začel razvijati leta 2005 kot naslednik prevajalnega sistema Pharoah. Nadaljnji razvoj projekta je sofinancirala EU v sklopu projektov EuroMatrix in EuroMatrixPlus. Moses je uporabljen na področju raziskovalnega dela in kot učni pripomoček za učenje o strojnem prevajanju. [6]

Moses poleg osnovnega strojnega prevajanja na osnovi fraz (PBMT), podpira tudi faktorsko prevajanje (Ang. Factored Translation), ki je nadgradnja PBMT, kjer se pri prevajanju uporabljajo dodatne jezikovne lastnosti. Primer dodatnih lastnosti so: besedne vrste, oseba, osnovna oblika, spol, čas, sklon in tako dalje.

Moses je izdelan predvsem z uporabo programskih jezikov C++ in Perl. Perl je uporabljen za orodja, namenjena predpripravi učnih podatkov, ter izdelavi prevajalnega modela. C++ pa je uporabljen za izdelavo sistema za prevajanje (Ang. decoder), ki s pomočjo statističnega prevajalnega modela ter vhodnih podatkov izdela prevod. Moses za izdelavo svojih modelov uporablja procesor za razliko od implementacij strojnega prevajanja na osnovi nevronskih mrež (NMT), ki uporabljajo za izdelavo lastnih modelov grafične kartice.

Izdelava prevajalnika s sistemom Moses potrebuje dve zunanji orodji, in sicer orodje za poravnavo besed in orodje za izdelavo statističnega jezikovnega modela. Uporabljeno orodje za poravnavo besed je običajno Giza++, vendar podpira tudi mgiza, ki je večnitna implementacija orodja Giza++, fast_align ter podobna orodja. Za izdelavo statističnega jezikovnega modela se lahko izbere med orodji: IRSTLM, SRILM, KenLM ter drugimi prostimi orodji (večina jih je tudi odprtokodnih) [11]. Sistem, ki bo

implementiran v zaključni nalogi, uporablja orodji Giza++ ter KenLM. Ti dve orodji bosta v nadaljevanju opisani.

3.1.1 KenLM

Kot je bilo predhodno omenjeno, obstaja več orodji za izdelavo statističnega jezikovnega modela, na primer IRSTLM, ki je bil izdelan v sklopu evropskega projekta EuroMatrixPlus, ali SRILM. Namesto teh dveh je bilo izbrano orodje KenLM, ki je bilo prvič predstavljen leta 2011. KenLM je odprtokodna implementacija orodja za izdelavo statističnega jezikovnega modela na podlagi n-terk. [7]

Glavna prednost KenLM pred preostalimi orodji je podpora večnitnega izvajanja. Na ta način se pospeši proces izdelave samega prevajalnika. Poleg hitre izdelave statističnega jezikovnega modela, model, izdelan s KenLM, zasede manj prostora na pomnilniku. V članku [7], v katerem je bil predstavljen statistični jezikovni model, avtor zapiše, da je KenLM $2,4$-krat hitrejši od SRILM ter pri tem uporabi zgolj 57% spomina, ki ga porabi SIRLM. Zaradi teh lastnosti je KenLM nadomestil IRSTLM kot osnovno orodje za izdelavo statističnih jezikovnih modelov v sistemu Moses.

3.1.2 Giza++

Za razliko od orodji za izdelavo statističnih jezikovnih modelov pri izbiri orodji za poravnavo besed ni velike izbire, na primer razvijalci Mosesa priporočajo ali uporabo fast_align ali eno od implementaciji orodja Giza. Orodje, ki je bilo izbrano v zaključnem delu, je Giza++ [6]. Giza++ je nadgradnja orodja iz leta 1999 Giza. Orodje je implementirano na osnovi IBM-modelov, ki so bili predstavljeni v poglavju 2.2.3 Poravnava besed.

Program Giza++ za svoje delovanje porabi največ procesorskega časa in prostora na pomnilniku pri izdelavi prevajalnega modela. Ta postopek lahko pospešimo tako, da izvedemo istočasno poravnavo besed iz italijanščine v slovenščino in iz slovaščine v italijanščino.

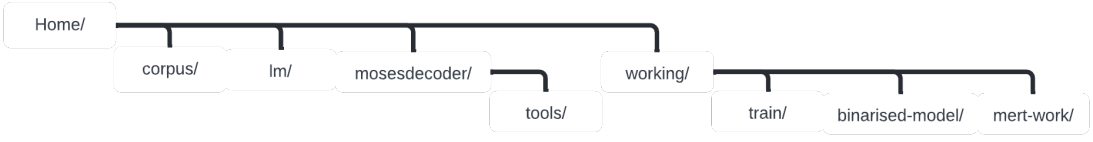
3.2 Izdelava

V tem podpoglavju je predstavljen postopek izdelave prevajalnika s pomočjo prej opisanih orodij. Poglavje je razdeljeno na več delov, v vsakem pa je opisan posamezni korak izdelave. Pred predstavitvijo samega postopka izdelave so predstavljeni korpusi, ki so uporabljeni v postopku izdelave in evalvacije prevajalnega sistema. Koraki izdelave so:

1. Predpriprava korpusov,
2. Izdelava jezikovnega modela,
3. Izdelava prevajalnega modela,
4. Izdelava jezikovnega modela,
5. Tunig,
6. Binarizacija.

Zatem bo predstavljena metoda za poganjanje prevajalnika. Poleg tega bo predstavljena tudi pomožna skripta, ki omogoča lažje poganjanje prevajalnika.

Implementacija uporablja datotečno strukturo, prikazano na Sliki 5. Mapa z imenom `Home` je domača mapa uporabnika. Mapa `corpus` vsebuje vse korpuse, ki so bili uporabljeni pri izdelavi prevajalnika. Mapa `lm` vsebuje statistični jezikovni model. V mapi `mosesdecoder` je shranjen sistem Moses. Mapa vsebuje tudi mapo `tools`, kjer so shranjena zunanja orodja, v tem primeru le Giza++. V mapi `working` pa so shranjeni delujoči prevajalni modeli. Mapa vsebuje tudi 3 druge mape: `train`, kjer je shranjen prevajalni model, `binarised-model`, v katerem je shranjen zadnji binariziran prevajalni model, ter `mert-work`, kjer so shranjene vse iteracije statističnega prevajalnega modela v fazi izboljšave modela.



Slika 5: Diagram datotečne strukture

Sistem, uporabljen za izdelavo prevajalnika, vsebuje procesor AMD Ryzen Threadripper 1950X, ki ima 16 jeder ter 32 niti. Sistem uporablja 32 GB delovni pomnilnik (RAM). Sistemu je dodeljena grafična kartica Nvidia GeForce GTX 1080 Ti, ki ni bila uporabljena, saj je Moses ne uporablja pri za izdelavi modelov.

3.2.1 Izbira korpusov

Izbira korpusov je pomemben del izdelave prevajalnika, ki temelji na SMT. Pomemben je zato, ker se s pomočjo korpusov izdelata statistični prevajalni in jezikovni model, ki sta osnovna elementa statističnega strojnega prevajanja. Zato je treba izbrati korpuse, ki so najbolj primerni za namen prevajalnika. Kriteriji, ki so bili uporabljeni, za izbiro korpusov, so: velikost, poravnanost ter domena korpusa. Pri tem so le bili izbrani dvojezični korpusi za jezikovni par slovenščina in italijanščina.

Velikost korpusa je pomembna, saj je potrebna različna količina podatkov v različnih fazah izdelave. Na primer korpus z veliko količino vnosov bo uporabljen pri izdelavi modelov, saj na ta način bo prevajalnik imel manj neznanih besed (Ang. Out-of-vocabular ali OOV) in bo tako lahko izdelal boljše prevode. Poravnanost korpusa je pomembna, saj če je vnos le v enem od jezikov, bo algoritem za poravnavo besed težje deloval in tako izdelal slabše ali celo napačne fraze ter na ta način znižal natančnost prevoda. Domena korpusa ne vpliva neposredno na kakovost prevoda, ampak na število OOV, ki jih bo imel prevajalnik. Na primer, če se želi izdelati prevajalnik, katerega domena je računalništvo, se ne bodo izbrali korpusi iz družboslovja, saj s tem izborom prevajalnik ne bo znal prevajati računalniških izrazov, ker jih ne pozna.

Korpusi so izbrani med tistimi, ki so na voljo v bazi OPUS. Namen projekta OPUS je zbiranje in distribucija prosto dostopnih dvojezičnih korpusov, ki so na voljno na spletu. OPUS omogoča iskanje korpusov za posamezni jezik oziroma jezikovni par. Za korpuse, ki so na voljo za izbran jezikovni par, so prikazani splošni podatki o posameznem korpusu. Na primer: število povedi, število žetonov za posamezni jezik ter različni formati korpusa. Format, primeren za uporabo z Mosesom, je imenovan Moses, na voljo pa so še v formatu XML (Ang. Extensible Markup Language) ter TMX (Ang. Translation memory exchange). Korpusi, ki so na voljo na OPUS, so pretežno iz treh domen: pravna oziroma administrativna besedila (zakonodaja EU), podnapisi filmov (podnapisi govorov TED) ter priročniki odprtokodnih sistemov (priročni za namizno okolje GNOME, priročni za namizno okolje KDE). [1]

Korpus v formatu, namenjenemu za uporabo v sistemu Moses, vsebuje dve datoteki, za vsak jezik posebej. Na primer korpus EUbookshop vsebuje dve datoteki, katerih ime se začne z `EUbookshop.it-sl` in ki imata končnico `.it` za Italijansko polovico korpusa ter `.sl` za slovensko.

Tabela 2: Primer vnosa povedi v korpusu

Italijanska poved	Perché è stato creato l'IPA:
Slovenska poved	Zakaj je bil instrument IPA vzpostavljen?

V Tabeli 2 je prikazana osma vrstica korpusa EUbookshop, kjer imata poved v italijanščini in poved v slovenščini isti pomen. Lastnost, da sta povedi z istim pomenom v isti vrstici, lahko uporabimo pri poravnavi besedil.

Korpusi, ki bodo uporabljeni pri izdelavi prevajalnega modela, so naslednji: TED2020, JRC-Acquis, EUbookshop. TED2020 je korpus, sestavljeni iz 4000 prepisov različnih govorov TED in TED-x. Te prepise so prevedli prostovoljcev v 100 različnih jezikov, med drugim tudi v slovenščino in italijanščino. Korpus je bil izdelan v sklopu članka [16] in objavljen na spletu 2. decembra 2020. Korpus je bil uporabljen pri procesu, imenovanem Tuning.

Korpus JRC-Acquis je sestavljen iz vseh evropskih zakonodaj in zakonodaj držav članic EU. Dokumenti, ki sestavljajo korpus, so bili izdani po letu 1950. Korpus je preveden v vse uradne jezike članic EU (23 različnih jezikov) do leta 2007. Sestavljen je iz dokumentov, ki so prevedeni v vsaj 10 evropskih jezikov [18]. Korpus je uporabljen za izračunanje metrike BLEU, katere postopek bo podrobneje opisan v poglavju 4 Evalvacija.

Zadnji korpus, ki je uporabljen, je EUbookshop. Sestavljen je iz različnih publikacij in knjig, ki so objavljene na spletni strani Urada za publikacije Evropske unije v sklopu publikacije EU (predhodno se je imenoval EUbookshop, od koder ime korpusa). Za izdelavo je bila uporabljena v2 različica korpusa, ki je bila objavljena 3. marca 2018. Korpus je bil uporabljen za izdelavo statističnega jezikovnega modela ter za izdelavo osnovnega statističnega prevajalnega modela [17].

3.2.2 Predpriprava korpusov

Po izbiri korpusov sledi izdelava prevajalnika. Prvi korak je predpriprava korpusov. Ta korak je potreben, saj prevajalnik za izdelavo statističnega jezikovnega in prevajalnega modela ne zna uporabljati povedi tako, kot so zapisane v korpusu. Zato je treba korpuse predpripraviti. Ta korak je v celoti narejen v mapi `~/corpus/` in poteka v treh fazah.

Prva faza je žetonizacija (Ang. tokenisation), ki jo izvedemo z naslednjim ukazom:

```
~/mosesdecoder/scripts/tokenizer/tokenizer.perl -l sl \
< ~/corpus/EUbookshop.it-sl.sl > \
~/corpus/EUbookshop.it-sl.tok.sl

~/mosesdecoder/scripts/tokenizer/tokenizer.perl -l it \
< ~/corpus/EUbookshop.it-sl.it > \
  ~/corpus/EUbookshop.it-sl.tok.it
```

Ukaz žetonizira vsak del korpusa, italijanski in slovenski, posebej. Jezik žetonizacije določimo tako, da je ukaz pognan z zastavico `-l` in pripisom `it` za italijanščino ter `sl` za slovenščino. Ukaz kot vhod prejme korpus v enem od jezikov ter vrne žetonizirano različico korpusa. Žetonizirana različica je shranjena v datoteki `ime korpusa.tok.*`, kjer je jezik korpusa določen s končnico (it za italijanščino, sl pa za slovenščino). V zgornjem primeru je uporabljen korpus EUbookshop, zato je žetonizirana datoteka za italijanski jezik shranjena v datoteki `EUbookshop.it-sl.tok.it`.

Tako je besedilo razdeljeno na žetone. Žeton je bodisi beseda bodisi ločilo, med seboj pa so ločeni s presledkom. Primer žetonizacije je prikazan na zgornjem delu Slike 6, kjer so naslednje slovenske povedi uporabljene kot primer:

- IPA - Instrument za predpristopno pomoč Nova razsežnost pomoči EU za širitev
- Zakaj je bil instrument IPA vzpostavljen?
- Kako instrument IPA deluje?

V podanem primeru so uporabljene zgolj slovenske povedi zaradi lažjega razumevanja primera, saj je isti postopek uporabljen tudi na italijanskem delu korpusa.

Naslednji korak je izbira najbolj primerne začetnice (Ang. truecasing). Ta korak je namenjen temu, da vse besed z veliko začetnico, ki niso lastna imena ali kratice, spremenimo v besede z malo začetnico. To pa zato, ker bo beseda najbolj verjetno uporabljena znotraj povedi in ne kot začetna beseda v povedi.

Postopek ima dva koraka. V prvem koraku je izdelan model za izbiro primerne začetnice. To je narejeno z naslednjim ukazom:

```
~/mosesdecoder/scripts/recaser/train-truecaser.perl --model \
~/corpus/truecase-model.sl --corpus \
~/corpus/EUbookshop.it-sl.tok.sl

~/mosesdecoder/scripts/recaser/train-truecaser.perl --model \
~/corpus/truecase-model.it --corpus \
~/corpus/EUbookshop.it-sl.tok.it
```

Ukaz izdela model za slovenski in italijanski jezik ločeno. Ukaz kot vhodni podatek prejme žetoniziran korpus ter vrne model, s katerim se bo vsem nadaljnjim korpusom določilo pravilno začetnico besed. Model je shranjen v datoteki `truecase-model.*`, kjer končnica pove jezik modela (it za italijanščino, sl pa za slovenščino). Model je izdelan zgolj enkrat ter ga uporabimo za vse korpuse.

Zdaj, ko je izdelan model za izbiro začetnice, ga lahko uporabimo na korpusu. To je narejeno z naslednjim ukazom:

```
~/mosesdecoder/scripts/recaser/truecase.perl --model \
~/corpus/truecase-model.sl < \
~/corpus/EUbookshop.it-sl.tok.sl > \
~/corpus/EUbookshop.it-sl.true.sl

~/mosesdecoder/scripts/recaser/truecase.perl --model \
~/corpus/truecase-model.it < \
~/corpus/EUbookshop.it-sl.tok.it > \
~/corpus/EUbookshop.it-sl.true.it
```

Ukaz spremeni začetnice besed v vsakem delu korpusa posebej. Ukaz prejme kot vhod model za izbiro začetnice in žetoniziran korpus ter vrne korpus z besedami, ki imajo primerno začetnico. Ta različica korpusa je shranjena v datoteki z imenom `ime korpusa.true.*`, kjer končnica datoteke pove jezik (it za italijanščino, sl pa za slovenščino). Na zgornjem primeru je uporabljen korpus EUbookshop, zato se datoteka, ki vsebuje žetone s pravilno začetnico za italijanski jezik, imenuje `EUbookshop.it-sl.true.it`.

Slika 6: Primer postopka predpriprave korpusov

Primer izbire pravilne začetnice je lahko viden v spodnjem delu Slike 6. Na sliki se besedam `zakaj in kako` spremeni začetnica. Primer je bil izdelan s podatki, pridobljenimi pri uporabi prej opisanega postopka, na korpusu EUbookshop.

Zadnji korak v predpripravi korpusov je čiščenje korpusa (Ang. cleaning). V tem koraku se izbrišejo vse vrstice v korpusu, ki lahko povzročijo težave pri izdelavi statističnega jezikovnega ali prevajalnega modela. Vrstice, ki so izbrisane, so bodisi prazne,

torej ne vsebujejo nobenega znaka, bodisi predolge, kar pomeni, da vsebujejo več besed, kot jih je želeno. V tem koraku so izbrisani tudi vsi odvečni presledki. To je storjeno z ukazom:

```
~/mosesdecoder/scripts/training/clean-corpus-n.perl \
~/corpus/EUbookshop.it-sl.true it sl \
~/corpus/EUbookshop.it-sl.clean 1 80
```

V tem primeru ukaz uredi oba dela korpusa hkrati. Ukaz sprejme kot vhod celoten korpus, ki so mu bile predhodno že določene pravilne začetnice, in najmanjše ter največje število besed v posamezni vrstici, in vrne prečiščen korpus. Ta različica korpusa je shranjena v datoteki z imenom `ime korpusa.clean.*`, pri čemer jezik je predstavljen s končnico (it za italijanščino, sl pa za slovenščino). Na zgornjem primeru je uporabljen korpus EUbookshop, zato je ime prečiščene datoteke za italijanski jezik `EUbookshop.it-sl.clean.it`.

3.2.3 Izdelava jezikovnega modela

Naslednji korak je izdelava modelov. Prvi model, ki bo izdelan, je statistični jezikovni model, ki bo uporabljen pri izdelavi statističnega prevajalnega modela. Za izdelavo statističnega jezikovnega modela, bo uporabljen slovenski del korpusa EUbookshop. Statistični jezikovni model bo izdelan zgolj za slovenski jezik, saj bo potreben pri izdelavi povedi v ciljnem jeziku, ki bo slovenščina. Model bo izdelan na podlagi trojk (3-terke). Celoten postopek izdelave jezikovnega modela bo potekl v mapi `~/lm/`. Model bo izdelan z uporabo naslednjega ukaza:

```
~/mosesdecoder/bin/lmplz -o 3 < \
~/corpus/EUbookshop.it-sl.true.sl > \
EUbookshop.it-sl.arpa.sl
```

Ukaz za izdelavo statističnega jezikovnega modela sprejme kot vhod slovenski korpus, ki so mu bile predhodno prirejene začetnice (žetoniziran, popravljene začetnice), vrne pa statistični jezikovni model. Model je sestavljen iz n delov. Število delov je odvisno od velikosti terk. Velikost terk je določeno z zastavico `-o`. Če je za primer vzet statistični jezikovni model, ki je uporabljen za izdelavo prevajalnika, se lahko predstavijo njegovi prvi štirje vpisi za različne terke s Tabelo 3. Statistični jezikovni model je shranjen v datoteko z imenom `ime modela.arpa.*`, pri čemer je jezik, za katerega je izdelan statistični jezikovni model, določen s končnico, v tem primeru je končnica `sl`, ki predstavlja slovenščino. Statistični jezikovni model, ki je uporabljen v prevajalniku, se imenuje `EUbookshop.it-sl.arpa.sl`.

Tabela 3: Primer jezikovnega modela

Posamične besede	<unk>	-6,3543544
	<s>	0
	</s>	-2,1062078
	instrument	-4,0474286
Pari besed	instrument </s>	-2,1602228
	za </s>	-2,4548802
	pomoč </s>	-1,7133037
	razsežnost </s>	-1,5847348
Trojkei besed	finančni instrument </s>	-2,3103042
	partnerski instrument </s>	-1,6179857
	načrtovalni instrument </s>	-0,66608083
	za za </s>	-2,5788026

V Tabeli 3 so prikazane posebne besede: <unk>, ki predstavlja neznano besedo (Ang. unknow, tudi uporabljena kratica OOV), <s> in </s>, ki pa predstavljata začetek in konec vrstice v XML notaciji. Nepozitivna števila, $\mathbb{R}_0^-$, ki so prikazana v zadnjem stolpcu, so verjetnosti posamezne terke. Verjetnost je po navadi prikazana kot vrednost iz intervala $[0, 1]$, v tem primeru pa je zapisana kot $\log_{10}$ verjetnosti. Na primer, za trojko »načrtovalni instrument </s>« je njena verjetnost $-0,66608083 = \log_{10} 0,215734285$.

Za pohitritev nalaganja jezikovnega modela v pomnilnik je treba statistični jezikovni model binarizirati. To je storjeno z ukazom:

```
~/mosesdecoder/bin/build_binary EUbookshop.it-sl.arpa.sl \
EUbookshop.it-sl.blm.sl
```

Ukaz prejme kot vhod statistični jezikovni model, ki je berljiv uporabniku, ter vrne binariziran model. Binariziran model je shranjen v datoteki z imenom `ime modela.blm.*`, kjer se s končnico določi jezik, za katerega je bil izdelan model, v tem primeru `sl` kot slovenščina, kot je lahko vidno iz imena statističnega jezikovnega modela, ki bo uporabljen pri izdelavi prevajalnika, `EUbookshop.it-sl.blm.sl`. Z binarizacijo modela se zmanjša velikost modela z 215MB na 131MB, kar predstavlja 39-% zmanjšanje zasedenega prostora na pomnilniku.

Statistični jezikovni model je lahko tudi uporabljen samostojno. V tem primeru

so podani: verjetnost posameznega žetona in celotne povedi, število žetonov, število OOV ter neodločenost modela za poved s ter brez OOV. V primeru povedi »Kaj je to slovenska poved?«, je to storjeno z ukazom `echo "Kaj je to slovenska poved ?" | ~/mosesdeco der/bin/query EUbookshop.it-sl.blm.sl`.

Ukaz izpiše, da je verjetnost povedi $-25,37917$, število žetonov je 7, število OOV je 0 ter nedoločenost je $4222,752597$.

3.2.4 Izdelava prevajalnega modela

Izdelava statističnega prevajalnega modela potrebuje prečiščen dvojezični korpus in statistični jezikovni model. Čeprav statistični prevajalni model po definiciji ne potrebuje statističnega jezikovnega modela za izdelavo, je v tem koraku ta potreben, saj se v tej točki ne izdela le statistični prevajalni model, ampak celotni prevajalni sistem. Izdelava modela poteka v mapi `~/working`. Model je izdelan z uporabo ukaza:

```
nohup nice ~/mosesdecoder/scripts/training/train-model.perl \
-cores 32 -root-dir train -corpus \
~/corpus/EUbookshop.it-sl.clean -f it -e sl \
-alignment grow-diag-final-and \
-reordering msd-bidirectional-fe \
-lm 0:3:/home/janisuban/lm/EUbookshop.it-sl.blm.sl:8 \
-external-bin-dir ~/mosesdecoder/tools >& training.out &
```

Ukaz, kot je bilo že prej omenjeno, sprejme dva vhodna podatka, ki sta: statistični jezikovni model in prečiščen dvojezični korpus, ter vrne datoteko s konfiguracijo sistema Moses, ki se imenuje `moses.ini`. Za pravilno in hitrejše delovanje ukaz uporablja različne zastavice. Zastavice so različnih vrst, nekatere določajo lokacijo določenih datotek (`-root-dir`, `-corpus`, `-lm`, `-external-bin-dir`), nekatere modele, ki bodo uporabljeni (`-alignment`, `-reordering`), nekatere pa lastnosti sistema (`-cores`). Zastavice, ki omogočajo upravljanje postoka izdelave sistema, so sledeče: `-lm`, `-alignment` in `-reordering`.

Zastavica `-lm` ne določa le poti do statističnega jezikovnega modela, ampak določa tudi faktor in dolžno terk. Oblika, v kateri je podan parameter zastavice, je <faktor>:<dolžina terke>:<ime jezikovnega modela>:<vrsta jezikovnega modela>. Pri izdelavi prevajalnika v zaključni nalogi je bil podan parameter `0:3:/home/janisuban/lm/EUbookshop .it-sl.blm.sl:8`, kjer 8 predstavlja KenLM.

Z zastavico `-alignment` določamo algoritem, ki bo uporabljen za izdelavo poravnave besed. Algoritem, ki je predstavljen v poglavju 2.2.3 Poravnava besed, je `grow-diag-final`. Za izdelavo prevajalnega sistema pa je uporabljen algoritem `grow-diag-final-and`, ki ima v zadnjem koraku, ki je prikazan na Algoritmu 1 med vrsti-

cama 9. in 12., *IN* namesto *ALI*, kar pomeni, da par dodamo le, če obe besedi še nista poravnani.

Zadnja zastavica, ki bo bo bolj predstavljena, je `-reordering`. Ta zastavica pove modelu, na kakšen način lahko spreminja vrstni red besed v povedi. Metoda, uporabljena v prevajalniku, je `msd-bidirectional-fe`, ki je razdeljena na tri dele in sicer na `msd`, `bidirectional` in `fe`. Del `msd` predstavlja različne orientacije, ki jih imajo besede, in sicer: monotone, swap in discontinuous. Beseda `bidirectional` predstavlja smer, na katero se modelira orientacija, ki se v tem primeru modelira v obe smeri, torej glede na predhodno in naslednjo frazo. Zadnji del, `fe`, pa predstavlja jezike, na katerih temelji preoblikovanje povedi. Ker je uporabljen `fe`, sta uporabljena oba jezika.

Ukaz potrebuje nekaj ur, da se izvršiti do konca na opisanem sistemu. Za pospešitev izvajanja je bilo podano ukazu s pomočjo zastavice `-cores 32` število niti, ki jih ima na razpolago za uporabo. Poleg tega je bila ukazu spremenjena vrednost niceness, ki vpliva na algoritem razvrščanja procesov, tako da ima proces več procesorskega časa. To je bilo storjeno s programom `nice`. Poleg tega je bil program pognan tudi z ukazom `nohup`, ki operacijskemu sistemu ne dovoli, da bi poslal procesu signal za prekinitev izvajanja.

Sedaj je prevajalnik za prevajanje iz italijanščine v slovenščino pripravljen za uporabo. Datoteka, kjer je shranjena konfiguracija sistema, je v mapi `~/working/train/`. Za prevajanje je uporabljen ukaz `echo` »poved, ki bo prevedena« `| ~/mosesdecoder/bin/moses -f ~/working/train/model/moses.ini`. V primeru povedi »Questo è solo un test di come funziona il traduttore.«, jo prevajalnik prevede v »To je samo, kako deluje Evropski prevajalce in pomočnike.«. Do napak v prevajanju pride zaradi nepoznavanja besed.

Trenutna verzija prevajalnika ni primerna za končno uporabo, saj se lahko izboljšata tako natančnost prevodov, kot čas, ki je potreben za prevod povedi. Izboljšava sistema bo predstavljena v poglavju 3.2.5 Tuning. Metoda pospešitve sistema pa bo predstavljena v poglavju 3.2.6 Binarizacija.

3.2.5 Tuning

V fazi, imenovani v angleščini Tuning, se izboljša natančnost modela. To je lahko storjeno, saj je statistični prevajalni model razdeljen na več poddelov, kot so na primer: model za preoblikovanje povedi, tabela fraz oziroma besed, statistični jezikovni model, saj je v tem primeru statistični prevajalni model celoten prevajalnik, in tako dalje. Natančnost je izboljšana tako, da se določijo uteži za posamezne poddele modela, za katere statističi prevajalni model izdela najboljši prevod v primerjavi z izhodiščnim prevodom iz korpusa. Metoda optimizacije, ki bo uporabljena v tej fazi, išče take

vrednosti uteži, ki izdelajo prevode z najmanjšim številom napak v prevodu. Metoda se v angleščini imenuje Minimum error rate training (MERT). Metoda uporablja različne metrike ali kombinacije metrik za izračun napake.

Za izboljšavo prevajalnika bo uporabljena, kot metrika za izračun napake, metrika BLEU, ki bo bolj podrobno opisana v poglavju 4.1.1 BLEU. Zgoraj opisani postopek se izvede z ukazom:

```
nohup nice ~/mosesdecoder/scripts/training/mert-moses.pl \
~/corpus/TED2020.it-sl.true.it \
~/corpus/TED2020.it-sl.true.sl \
~/mosesdecoder/bin/moses train/model/moses.ini --mertdir \
~/mosesdecoder/bin/ --decoder-flags="-threads 32" \
&> mert.out &
```

Ukaz prejme kot vhod korpus, ki bo uporabljen za optimizacijo uteži. Korpus ima izbrane primerne začetnice besed, vsak del korpusa pa je podan ločeno. Ukaz prejme kot vhod tudi statistični prevajalni model, ki bo izboljšan, in vrne model s popravljenimi utežmi.

Ta korak izdelave prevajalnika je najbolj počasen. Za razliko od izdelave statističnega prevajalnega modela, ki potrebuje nekaj ur, ta korak potrebuje četrtino dneva, da se izvede do konca. Pri tem je ukaz imel na razpolago vseh 32 niti ter je bil pognan z ukazom `nohup nice`. Razlog za tako dolgotrajno izvajanje je večkratno prevajanje celotnega korpusa in preračunavanja uteži, v tem primeru je to bilo storjeno 6-krat.

Tabela 4: Primerjave uteži prevajalnika pred in po fazi Tuning

Ime uteži	Pred	Po
UnknownWordPenalty0	1	1
WordPenalty0	-1	-0,188909
PhrasePenalty0	0,2	0,135477
TranslationModel0	0,2 0,2 0,2 0,2	0,0382576 0,0531127 0,0471983 0,0471983
LexicalReordering0	0,3 0,3 0,3 0,3 0,3 0,3	0,0707974 0,0597219 0,0699762 0,0663338 0,00386846 0,0707974
Distortion0	0,3	0,0707974
LM0	0,5	0,0775543

Če se primerja datoteko `moses.ini`, ki je bila podana kot osnova v fazi tuning, s tisto, pridobljeno v tej fazi, se opazi, da so vse uteži bile spremenjene. Spremembe so prikazane v Tabeli 4. Največja sprememba je vidna pri utežeh `TranslationModel0`, ki so bile predhodno vse vrednosti $0,2$, in `LexicalReordering0`, ki pa so bile predhodno vse vrednosti $0,3$, zdaj pa imata oba modela različne vrednosti.

3.2.6 Binarizacija

Druga izboljšava, ki se lahko naredi, ne izboljša kakovosti prevoda, ampak zmanjša čas, ki ga prevajalnik potrebuje za izdelavo prevoda. To je storjeno tako, da se stisne (Ang. compress) tabelo fraz in leksikografsko tabelo. Na ta način datoteke porabijo manj prostora na pomnilniku, bodisi na zunanjem pomnilniku bodisi na delovnem pomnilniku. Tako je za prenos tabel iz zunanjega na delovni pomnilnik potrebno manj časa.

Za pravilno delovanje te faze izdelave se mora sistem Moses prevesti z dodano knjižnico CMPH (Ang. C Minimal Perfect Hashing), ki omogoča dobro stiskanje podatkov ter hitri poizvedovalni čas. Postopek binarizacije (Ang. binarise) je storjen še vedno v mapi `~/working/`, in sicer s sledečima ukazoma:

```
~/mosesdecoder/bin/processPhraseTableMin -in \
train/model/phrase-table.gz -nscores 4 -out \
binarised-model/phrase-table

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

Zgornji ukaz binarizira tabele fraz, spodnji pa leksikografske tabele. Oba ukaza sprejmeta kot vhod tabelo, ki jo bosta binarizirala, ter vrneta binarizirano tabelo. Pri binarizaciji tabele fraz je treba navesti, koliko rezultatov želimo shraniti, v tem primeru so bili shranjenih štirje rezultati. Če želimo tabeli uporabiti za prevajanje, je treba spremeniti lokacijo tabel v datoteki `moses.ini` tako, da sta uporabljeni binarizirani različici.

Tabela 5: Primerjava časa prevajanja pred in po binarizaciji

	Pred	Po
Nalaganje tabele fraz	156,94 s	/
Izdelava vhodno-izhodnega objekta	299,92 s	0,37 s
Prevajanje	0,02 s	0,04 s
Končni čas	310,11 s	0,44 s

Za primer, ki je predstavljen v Tabeli 5, je bil analiziran čas prevoda povedi »Che tempo fa oggi?«. Časi so bili pridobljeni iz poročila, ki ga izpiše sistem Moses na standardni izhod za napake (Ang. standard error). Za nebinariziran prevod je bil uporabljen model pred fazo Tuninga, za binarizran prevod pa model po fazi Tuninga. To je bilo storjeno, saj se ne primerja kakovosti prevoda, ampak le čas, ki ga potrebuje prevajalnik za prevod povedi. Pospešitev procesa prevajanja je 705-kratna, saj sistem potrebuje večino časa za premik tabele v delovni pomnilnik. Čeprav je prevajalnik pred binarizacijo za sam prevod potreboval manj časa, je celoten postopek prevajanja bil daljši, in zato je vseeno bolje binarizirat tabeli.

3.2.7 Poganjanje prevajalnika

Sedaj se prevajalnik lahko uporablja. Lahko se poganja z ukazom `echo` »poved, ki bo prevedena« `| ~/mosesdecoder/bin/moses -f ~/working/binarised-model/moses.ini`, pri čemer je treba vhodno poved predhodno žetonizirati, izhodno poved pa naknadno razžetonizirti (Ang. detokenize). Moses izpisuje pri prevajanju trenutno stanje na standardni izhod za napake. Zato je bila izdelana naslednja skripta:

```
#!/bin/bash
# -p      izdela datoteko z prevdom
# -t n    izdelava n. testnega primera
DATE=$(date +'%F-%H-%M-%S')
IN1=$1
IN=$(echo $IN1 | \
~/mosesdecoder/scripts/tokenizer/tokenizer.perl -l it \
2> /dev/null)
OUT1=$(echo $IN | /home/janisuban/mosesdecoder/bin/moses -f \
/home/janisuban/working/binarised-model/moses.ini \
2> /dev/null)
OUT=$(echo $OUT1 |\
~/mosesdecoder/scripts/tokenizer/detokenizer.perl -l sl \
2> /dev/null)
if [ "$2" == "-p" ]
then
echo $IN1>Prevod$DATE.txt
echo $OUT>>Prevod$DATE.txt
fi
echo $OUT
```

Glavni namen te skripte je poenostaviti uporabo prevajalnika. Skripta samodejno žetonizira besedilo, ki ga želimo prevesti, nato žetone prevede injih na koncu sestavi skupaj. Druga poenostavitev uporabe prevajalnika je dosežena tako, da je celotni standardni izhod za napake preusmerjen na izhod `/dev/null`. Na ta način se izpiše le prevod povedi.

Skripta omogoča uporabo dveh zastavic, in sicer `-p` in `-t`. Naenkrat lahko izberemo le eno zastavico ter zastavico postavimo za besedilom, ki ga želimo prevesti. Zastavica `-p` izdela datoteko, v kateri sta shranjeni obe povedi. V prvi vrstici je zapisana izvorna poved v italijanščini, v drugi vrstici pa prevod te povedi v slovenščini. Ime datoteke, v katero se shrani prevod, je oblike `Prevod%F-%H-%M-%S.txt`, pri čemer `%F` predstavlja datum v obliki leto, mesec, dan, ki jih med seboj loči s pomišljajem, `%H` predstavlja uro, `%M` minute ter `%S` sekunde. Primer imena datoteke, ki ga izdela ta ukaz, je `Prevod2022-04-27-12-30-00.txt`.

Zastavica `-t` ni implementirana v zgornji skripti, saj je njen namen je izdelava testnih primerov, ki bodo uporabljeni v poglavju 4.2 Primerjava, in zato bo njena implementacija predstavljena takrat, ko bo uporabljena. Podobno kot zastavica `-p`, bosta izvorna italijanska poved in prevedena poved v slovenščino zapisani v datoteko, pri čemer bo pred izvorno povedjo zapisana še številka testnega primera, ki bo podana takoj za zastavico, povedi pa bodo zapisane v datoteko `Test.txt`. Glavna razlika med zastavicama je, da zastavica `-p` vedno izdela novo datoteko ali jo prepiše, če že obstaja, saj je namenjena izdelavi datoteke le za en prevod. Zastavica `-t` pa dopiše prevode na konec datoteke, če datoteka že obstaja, saj je njen namen izdelati datoteko z vsemi testnimi primeri.

4 Evalvacija

V tem poglavju bo analizirana učinkovitost izdelanega prevajalnika v primerjavi z že obstoječimi. Poglavje je razdeljeno na dva dela. V prvem delu sta predstavljeni metodi evalvacije prevajalnikov: BLEU (avtomatizirana) in človeška ocena.

Drugi del pa temelji na medsebojni primerjavi prevajalnikov: Googlovim Prevajalnikom, prevajalnikom podjetja DeepL in prevajalnikom, katerega postopek izdelave je bil predstavljen v poglavju 3 Implementacija, ter človeškim prevodom.

4.1 Metode evalvacije

Ocena kakovosti prevodov, ki jih izdela prevajalnik, ali primerjava prevajalnikov z drugimi prevajalniki je izdelana s pomočjo ene ali več metod evalvacije. Metode evalvacije so metrike, lestvice, pristopi evalvacije, ki omogočajo lažjo medsebojno primerjavo prevajalnikov. Poznamo dva pristopa do evalvacije prevajalnikov. Prvi pristop je avtomatizirana evalvacija prevajalnika, pri čemer ni potrebna človeška pomoč pri evalvaciji. Primer takega pristopa je metrika BLEU, ki bo uporabljena pri evalvaciji prevajalnika.

Drugi pristop k evalvaciji prevajalnika temelji na človeškem mnenju o prevodu. Pri tem se uporablja predhodno definirana lestvica, ki določa mnenje o kakovosti prevoda. Ta metoda bo uporabljena za primerjavo med prevajalniki.

4.1.1 BLEU

Prva predstavljena in uporabljena metoda evalvacije prevajalnika je metrika, imenovana v angleščini Bilingual Evaluation Understudy (BLEU). BLEU je metoda evalvacije, pri čemer se prevod primerja z referenčnim besedilom. Prevod se ovrednoti glede na število besed oziroma terk besed, ki se ujemajo z referenčnim besedilom. Končna ocena je povprečje vseh ocen referenčnih besedil v korpusu. Metrika BLEU je prva metoda ocenjevanja prevajalnikov, ki je prikazala visoko korelacijo med samo metriko in človeško evalvacijo. V članku [5], kjer je bila predstavljena metrika BLEU, je korelacijski koeficient med človeško evalvacijo in rezultatom metrike BLEU $0,96$. Metrika BLEU vrne vedno rezultat iz intervala $[0, 1]$. Za lažjo berljivost bodo rezultati metrike pomnoženi s 100.

Metrika BLEU meri kakovost besedila le glede na referenčni prevod, kar je lahko velika težava. Zato metrika slabše oceni prevod, ki ne uporablja istih besed, kot so v referenčnem prevodu, temveč sopomenke. Metrika ne upošteva jezikovno pravilne sestave povedi, zato bo najbolje ocenjen prevod, ki je permutacija referenčnega prevoda. S časom je metrika postala vedno manj relevantna, ampak bo kljub temu uporabljena v tej evalvaciji, saj je preprosta za razumevanje.

Za izračun metrike BLEU je bil uporabljen korpus JRC-Acquis, ki je bil predpripravljen vse do odstranitve povedi (žetonizacija in izbira pravilne začetnice). Za tem je bil filtriran statistični prevajalni model, saj se na ta način pospeši prevajanje korpusa. S filtriranjem se odstranijo iz modela vsa vrjetnosti, ki niso potrebne za prevod korpusa. To je storjeno z ukazom:

```
~/mosesdecoder/scripts/training/filter-model-given-input.pl \
JRC-Acquis mert-work/moses.ini \
~/corpus/JRC-Acquis.it-sl.true.it -Binarizer \
~/mosesdecoder/bin/processPhraseTableMin
```

Za hitrejšo izdelavo prevodov, ki bodo uporabljeni za izračun metrike, je bil uporabljen binariziran model. Zato je treba na vhod podati še lokacijo binarizirane tabele fraz. To je storjeno z zastavico `-Binarizer`, za njo pa je navedena lokacija tabele. Naslednji korak je prevod korpusa, ki je storjen z ukazom:

```
nohup nice ~/mosesdecoder/bin/moses -f \
~/working/JRC-Acquis/moses.ini \
< ~/corpus/JRC-Acquis.it-sl.true.it \
> ~/working/JRC-Acquis.it-sl.tratranslated.sl \
2> ~/working/JRC-Acquis.it-sl.out &
```

Za prevod sta uporabljena predhodno filtriran statistični prevajalni model in korpus, s katerim je bil filtriran statistični prevajalni model, pri čemer je podan zgolj italijanski del, kajti slovenskega bo uporabljen za izračun metrike BLEU. Pri tem je standardni izhod preusmerjen v datoteko, imenovano `JRC-Acquis.it-sl.tratranslated.sl`, standardni izhod za napake pa v datoteko imenovano `JRC-Acquis.it-sl.out`. Ker je potek prevajanja počasen, je bil program pognan z ukazoma `nohup nice`, ki preprečita pošiljanje signala za prekinitev procesa (`nohup`) ter omogočata višjo prednost izvajanja procesa (`nice`).

Zdaj, ko je izdelan prevod korpusa v slovenščino, se lahko izračuna metrika BLEU. Program, ki bo izračunal vrednost metrike, je že del Mosesa, in se požene z ukazom:

```
~/mosesdecoder/scripts/generic/multi-bleu.perl \
-lc ~/corpus/JRC-Acquis.it-sl.true.sl \
< ~/working/JRC-Acquis.it-sl.tratranslated.sl > BLEU.out
```

Ukaz sprejme kot vhod originali slovenski del korpusa, ki je podan kot parameter, ter prevedeni v slovaščino italijanski del korpusa, ki je podan s preusmeritvijo standardnega vhoda iz predhodno izdelane datoteke. Rezultat je izračunana vrednost metrike BLEU, ki je podana na standardnem izhodu in je preusmerjena v datoteko `BLEU.out`.

Pred predstavitvijo vrednosti metrike BLEU za izdelan prevajalnik je treba podati razlago o interpretaciji rezultatov. V članku [12] je predstavljena najbolj razširjena interpretacija metrike BLEU. To interpretacijo uporablja Google v svojih navodilih za izdelavo prevajalnika s pomočjo GNMT (Ang. Google Neural Machine Translation).

Vrednosti BLEU nad 30 se lahko interpretirajo kot prevajalnik izdeluje dobre razumljive prevode. Za vrednosti BLEU nad 50 pa se lahko razume, da so prevodi kakovostni ter zvenijo naravno. Pri tem je pomembno omeniti, da rezultat BLEU ni samo odvisen od prevajalnika, ki je uporabljen, ampak tudi od jezikovnega para, korpusa, ki je uporabljen, ter velikost korpusa, saj če je uporabljen večji korpus, se bo izboljšala natančnost metrike. Če se želi dva prevajalnika primerjati z uporabo metrike BLEU, je treba uporabiti isti korpus pri obeh prevajalnikih. Pri tem velja, da višji kot je rezultat, boljši je prevajalni sistem.

Rezultat metrike BLEU, izračunane z uporabo korpusa JRC-Acquis, je $28,15$. To je lepo vidno na primeru italijanske povedi »Questo è solo un test di come funziona il traduttore«, ki bi jo človek prevedel v slovensko poved »To je samo test delovanja prevajalnika«. Prevajalnik pa to isto poved prevede v »To je samo, kako deluje Evropski prevajalce in pomočnike«, pri čemer napačno prevede besedo prevajalnik ter uporabi odvisni stavek, ki ga človek ne bi. Poleg tega prevajalnik vsebuje veliko OOV, kar zniža kakovost prevodov.

4.1.2 Človeška ocena prevoda

Druga metoda evalvacije prevajalnika, ki bo uporabljena, je človeška ocena prevoda. Kot pove samo ime metode, ta temelji na ideji, da ljudje, ki govorijo oba jezika ali pa vsaj ciljni jezik, v našem primeru slovenščino, ocenijo kakovost prevoda. Pri tem obstaja več različnih lestvic ocenjevanja, ki se med seboj razlikujejo po številu stopenj lestvice ter samem številu lestvic.

Različni avtorji uporabljajo lestvice z različnim številom stopenj, in sicer petstopenske, šeststopenske ali sedemstopenske, ki so najbolj uporabljene. Pri tem se uporabljajo različne začetne in končne vrednosti. Najbolj pogosto se uporablja lestvice med 1 in, recimo, 5, če se vzame za primer petstopensko lestvico, ponekod pa se uporablja tudi lestvice med 0 in, recimo, 4, če se vzame ponovno petstopensko lestvico. Število lestvic, ki se uporabljajo za evalvacijo prevajalnika, je lahko 1 ali 2. To je odvisno odtega, ali se kakovost prevoda deli ali se ne deli na jezikovni in vsebinski kakovosti.

Lestvica, ki bo uporabljena za evalvacijo in primerjavo prevajalnikov, je petstopenska, z ocenami med 1 in 5, pri čemer bosta hkrati evalvirani jezikovni in vsebinski kakovosti. Lestvica, ki bo uporabljena, je naslednja:

1. Poved je nejasna in brez pomena.
2. Jezik povedi je nejasen, a se vseeno da razumeti njen pomen.
3. Prevod je razumljiv z večjimi slovničnimi napakami.
4. Prevod je razumljiv z manjšimi slovničnimi napakami.
5. Prevod je razumljiv z minimalnimi slovničnimi napakami.

Evalvacija bo potekala tako, da bodo osebi prikazane referenčne povedi v italijanščini iz Tabele 6 ter njeni prevodi v slovenščino. Oseba bo morala te prevode oceniti z zgoraj predstavljeno lestvico.

Tabela 6: Referenčne povedi v italijanščini

Številka primera	Italijanske povedi
1.	Il tempo è bello, usciamo oggi pomeriggio?
2.	La Commissione Europea delibera sulle nuove leggi.
3.	Novak è il più frequente cognome in Slovenia.
4.	In primavera la maggior parte delle piante fiorisce.
5.	Come va con la scrittura della tesi di laurea?

Rezultati te evalvacije bodo predstavljeni v poglavju 4.2 Primerjava z drugimi prevajalniki, saj bodo iste referenčne povedi uporabljene pri medsebojni primerjavi prevajalnih sistemov.

4.2 Primerjava z drugimi prevajalniki

V tem poglavju bosta predstavljena postopek medsebojne primerjave prevajalnikov in analiza rezultatov primerjave. Prevajalniki, ki bodo medsebojno primerjani, so: Google Prevajalnik, DeepL in prevajalnik, izdelan v sklopu zaključne naloge. Poleg teh treh prevajalnikov bomo primerjali tudi človeški prevod.

Za primerjavo bom uporabil referenčne povedi, ki so navedeni v Tabeli 6. Prevod referenčnih povedi z različnimi prevajalniki in človeški prevod so prikazani v Tabeli 7.

Tabela 7: Prevodi uporabljeni za testiranje

	Google prevajlanik	DeepL	Implementiran prevajalnik	Človeški prevod
1.	Vreme je lepo, gremo popoldne ven?	Vreme je lepo, gremo popoldne ven?	Evropski čas je čudovito, usciamo danes popoldne?	Gremo danes popoldne ven, saj je lepo vreme?
2.	Evropska komisija razpravlja o novih zakonih.	Evropska komisija razpravlja o novih zakonih.	Evropska komisija odloča o novih zakonov.	Evropska komisija odloča o novih zakonih.
3.	Novak je najpogostejši priimek v Sloveniji.	Novak je najpogostejši priimek v Sloveniji.	Novak je najpogostejša priimek v Sloveniji.	Novak je najpogostejši priimek v Sloveniji.
4.	Spomladi večina rastlin zacveti.	Spomladi cveti večina rastlin.	V pomladi večina rastlin cveti.	Spomladi večina rastlin cveti.
5.	Kako ste s pisanjem diplomske naloge?	Kako poteka pisanje vaše diplomske naloge?	Kot je treba s znanja in trditev diplomo?	Kako gre pisanje diplomske naloge?

Prevodi, ki jih je izdelal človek, so bili v Tabeli 7 izdelani sočasno s testnimi primeri v italijanščini. Prevodi z Google Prevajalnikom in DeepL-jem so bili izdelani ročno, saj za avtomatizirano prevajanje z uporabo aplikacijskega programskega vmesnika (Ang. Application programming interface ali API) zahteva registracijo v sistem (DeepL) ali pa je storitev plačljiva (Google Prevajalnik prek Google Cloud, pri čemer se ob registraciji pridobi 300 ameriških dolarjev dobroimetja za testiranje platforme).

Za izdelavo testnih prevodov v Tabeli 7 s prevajalnikom, čigav postopek izdelave je bil opisan v zaključni nalogi, pa je bila skripta za poganjanje prevajalnika popravljena, tako da omogoča avtomatizirano prevajanje testnih primerov. To je bilo storjeno z dopolnitvijo skripte z naslednjim delom:

```
...
OUT=$(echo $OUT1 | \
~/mosesdecoder/scripts/tokenizer/detokenizer.perl -l sl \
2> /dev/null)
if [ "$2" == "-t" ]
then
echo $3>>Test.txt
echo $OUT>>Test.txt
echo "">>Test.txt
exit 0
fi
if [ "$2" == "-p" ]
...
```

Če se zdaj analizirajo prevodi, ki so bili izdelani s prevajalnikom, opisanim v zaključni nalogi (v četrtem stolpcu Tabele 7), se lahko opazita dve neznani besedi v prevajalniku (OOV), in sicer priimek Novak in besedo usciamo. Želimo, da prevajalnik ne prevede priimkov in imen, saj se imen in priimkov oseb ne prevaja. Beseda

usciamo pa bi se morala prevesti v besedo zvezo gremo ven, kot se to vidi pri ostalih treh prevodih. Poleg OOV ima prevajalnik veliko napačno prevedenih besed, ki nimajo istega pomena kot v italijanščini. To se vidi v prevodu pete referenčne povedi, kjer je edina pravilno prevedena beseda je diploma (laurea), ter v prvi referenčni povedi, kjer je beseda il tempo prevedena v evropski čas namesto v besedo vreme.

Poleg prevajalnih napak se lahko opazijo tudi slovnične napake, če se osredotočimo zgolj na vsebinsko pravilno prevedene povedi. V povedi številka dva se lahko opazi neujemanje v sklonu pri besedni zvezi o novih zakonov, kjer bi moral biti uporabljen mestnik (o novih zakonih), je uporabljen rodilnik. Druga slovnična napaka pa je besedna zveza najpogostejša priimek, kjer se ne ujemata spola med pridevnikom (ženski) in samostalnikom (moški). Pravilen prevod bi bil najpogostejši priimek.

Mnenje ljudi o prevajalnikih je bilo zbrano s pomočjo ankete, ki je bila izdelana s slovenskim odprtokodnim orodjem 1KA[2]. V anketi so uporabniku predstavljeni izvirna poved v italijanščini in štirje prevodi v slovenščino. Anketiranec mora vsak prevod oceniti z lestvico, opisano v poglavju 4.1.2 Človeška ocena prevoda. Na koncu pa je anketiranec vprašan po nivo uporabe slovenščine in italijanščine (materni jezik, jezik okolja ali tuji jezik), starosti in spolu.

4.2.1 Analiza rezultatov

Anketo je izpolnilo 15 oseb, in sicer 10 moških in 5 žensk. Največ ljudi je bilo starih med 20 in 24 leti, kot je to razvidno iz Tabele 8. Opazi se lahko tudi, da je bilo 5 oseb starejših od 35 let, 2 v starostni skupini od 35 do 39 in 3 v starostni skupini več kot 39.

Tabela 8: Število udeležencev po starostnih skupinah

<15	15–19	20–24	25–29	30–34	35–39	>39
0	3	6	1	0	2	3

Iz Tabele 9 se lahko opazi, da je pri dveh osebah italijanščina materni jezik slovenščina pa jezik okolja, kar pomeni, da sta najverjetneje ti dve osebi predstavnika italijanske manjšine. Večini anketirancev je slovenščina materni jezik (13 ljudem). Čeprav večina ljudi, ki govori oba jezika, je iz slovenske Istre, kjer je italijanščina jezik okolja za slovensko govoreče prebivalstvo, je 5 oseb odgovorilo, da je italijanščina zanje tuji jezik. Ker preostali pari nivojev znanja italijanščine in slovenščine, ki niso zapisani v Tabeli 9, nimajo nobenega udeleženca v anketi s tako ravnjo znanja, tudi niso bili zapisani.

[2]https://1ka.arnes.si/a/44821e92

Tabela 9: Število udeležencev za različne pare znanja italijanščine in slovenščine

Nivo italijanščine	Nivo slovenščine	Število ljudi
Materni jezik	Jezik okolja	2
Jezik okolja	Materni jezik	8
Tuji jezik	Materni jezik	5

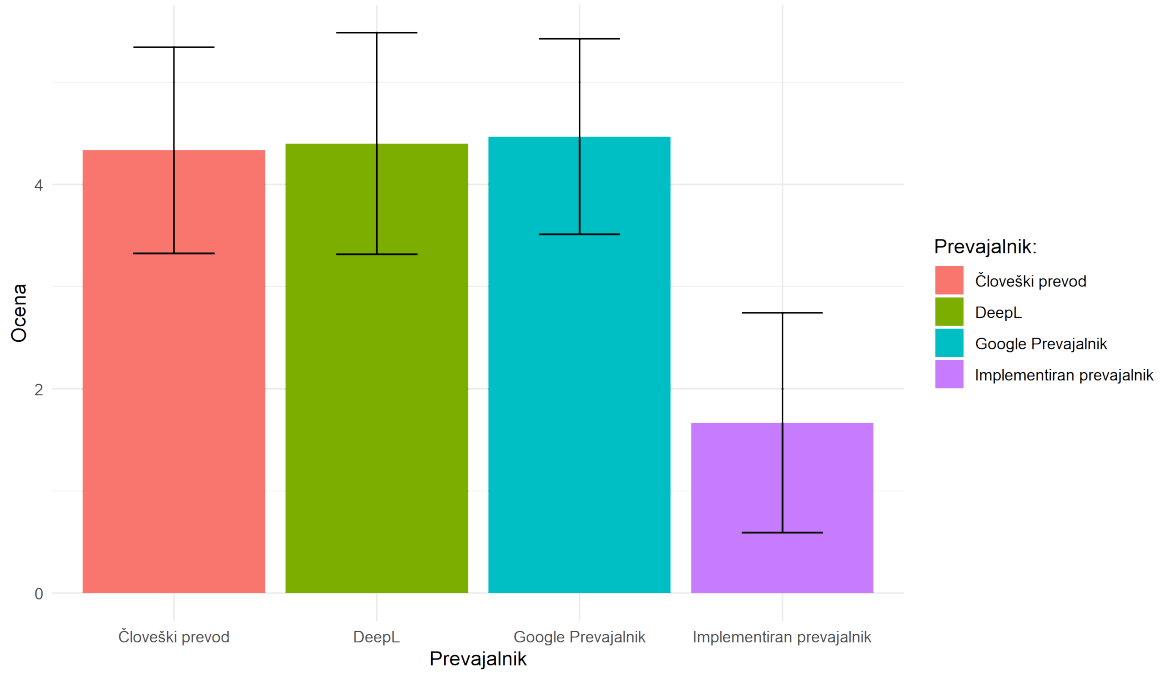
Mnenje anketirancev o prevodih referenčne povedi številka 1, ki so bili izdelani z Google Prevajalnikom, DeepL in človeškim prevodom, so najboljše izdelani prevodi za to referenčno poved, kot se to lahko vidi iz Tabele 10, kjer so si vrednosti mediane, modusa in aritmetične sredine med seboj podobne.

Tabela 10: Aritmetična sredina, standardni odklon, mediana in modus za 1. referenčno poved

Prevajalnik	Aritmetična sredina	Standardni odklon	Mediana	Modus
Google Prevajalnik	4,467	0,957	5	5
DeepL	4,400	1,083	5	5
Implementiran prevajalnik	1,667	1,075	1	1
Človeški prevod	4,333	1,011	5	5

Rezultati iz Tabele 10 so prikazani tudi grafično na Sliki 7. Vrednosti, ki sta predstavljene na grafu, sta: povprečna ocena prevoda (aritmetična sredina) in interval, na katerem leži 68 % ocen.

Razlog za tako nizko oceno prevajalnika, ki je bil implementiran v sklopu zaključne naloge, je najverjetneje veliko število neznanih besed (OOV). V tem primeru je to beseda usciamo, ter odkritje napačnih vzorcev, kot je na primer evropski čas za besedo tempo. Čeprav je 10 ljudi ocenilo poved kot nerazumljiv, je še vedno 5 ljudi ocenilo, da je pomen povedi razumljiv. Od teh 5 oseb 2 osebi menita, da ima poved ima le manjše jezikovne napake.



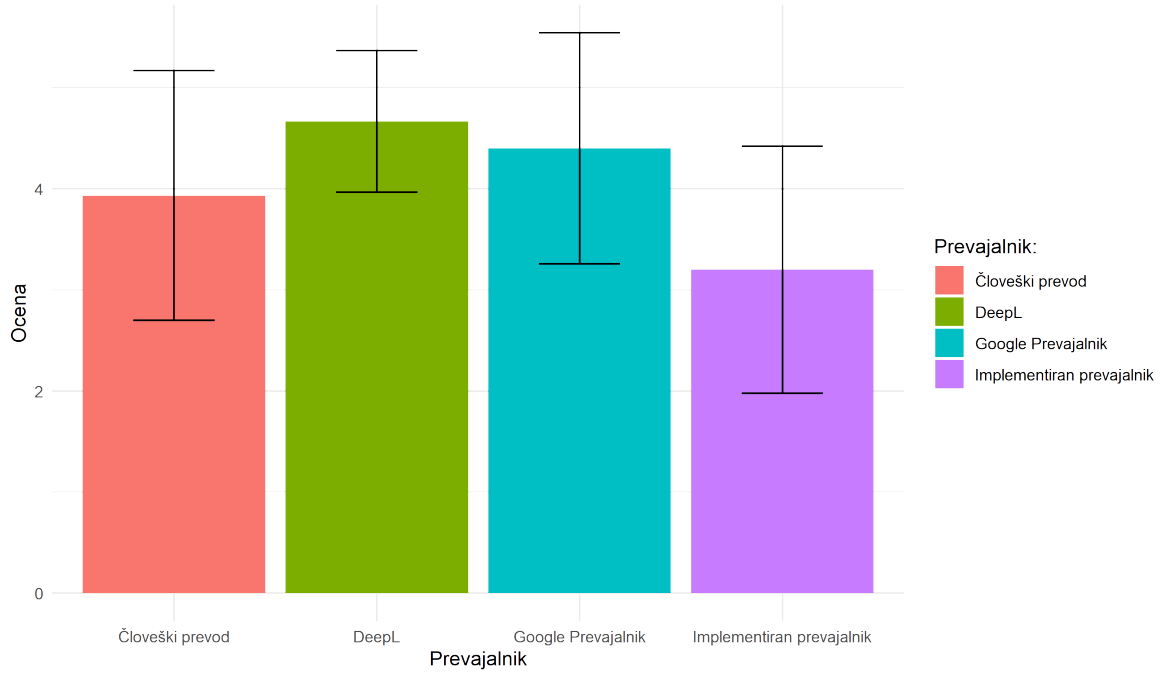
Slika 7: Graf s povprečnimi ocenami 1. referenčne povedi

Za prevode referenčne povedi številka 2 anketiranci menijo, da so najboljši prevodi izdelani z Google Prevajalnikom in DeepL, pri čemer se človeški prevod približuje njunim rezultatom. To lahko opazimo iz modusov ter median, ki so predstavljeni v Tabeli 11. Možen razlog za tako nizko povprečno oceno človeškega prevoda je uporaba sopomenke, v tem primeru sta to besedi: razpravlja in odloča.

Tabela 11: Aritmetična sredina, standardni odklon, mediana in modus za 2. referenčno poved

Prevajalnik	Aritmetična sredina	Standardni odklon	Mediana	Modus
Google Prevajalnik	4,400	1,143	5	5
DeepL	4,667	0,699	5	5
Implementiran prevajalnik	3,200	1,222	3	4
Človeški prevod	3.933	1.236	4	5

V tem primeru je prevajalnik, ki je bil izdelan v sklopu zaključne naloge, izdelal boljši prevod kot v predhodnem primeru. To se lahko vidi iz modusa in mediane v Tabeli 11 ali pa iz grafa, ki je prikazan na Sliki 8.



Slika 8: Graf s povprečnimi ocenami 2. referenčne povedi

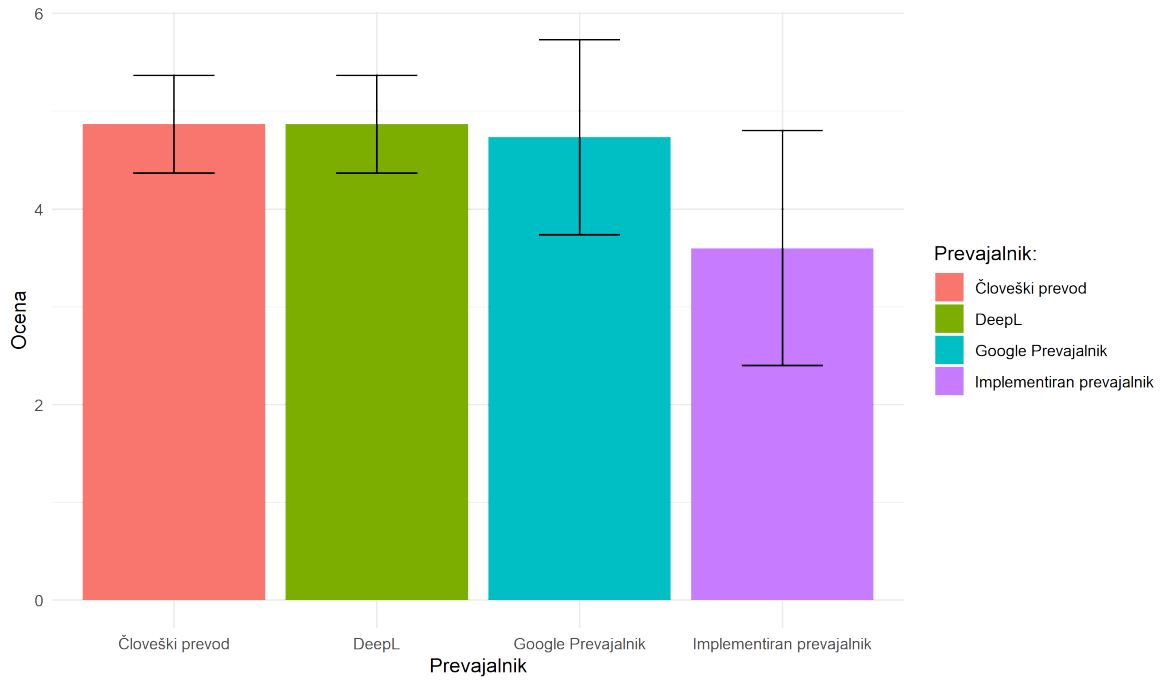
Možen razlog za tako visoko oceno prevoda, izdelanega s prevajalnikom, ki je bil implementiran v zaključni nalogi, je domena uporabljenega korpusa pri izdelavi, evropske publikacije objavljene do leta 2018. Zato je bila tudi izbrana ta referenčna poved, saj omogoča testiranje prevajalnika s povedjo iz domene korpusa.

Prevoda referenčne povedi številka 3, izdelana z Google Prevajalnikom, DeepL ter človeškim prevodom, so bili ocenjeni medseboj zelo podobno. To se lahko razbere iz Tabele 12. Poleg tega se lahko opazi, da so anketiranci ocenili prevod, izdelan s prevajalnikom, predstavljenim v zaključni nalogi, s 4, bodisi mediana bodisi modus.

Tabela 12: Aritmetična sredina, standardni odklon, mediana in modus za 3. referenčno poved

Prevajalnik	Aritmetična sredina	Standardni odklon	Mediana	Modus
Google Prevajalnik	4,733	0,998	5	5
DeepL	4,867	0,499	5	5
Implementiran prevajalnik	3,600	1,200	4	4
Človeški prevod	4,867	0,499	5	5

Na grafu, ki je prikazan na Sliki 9, je možno opaziti iste ugotovitve. Razlog za tako oceno prevajalnika, ki je bil izdelan v zaključni nalogi, je zgolj manjša jezikovna napaka. Prevajalnik je uporabil preseżnik v ženskem spolu namesto v moškem, saj je samostalnik priimek moškega spola.



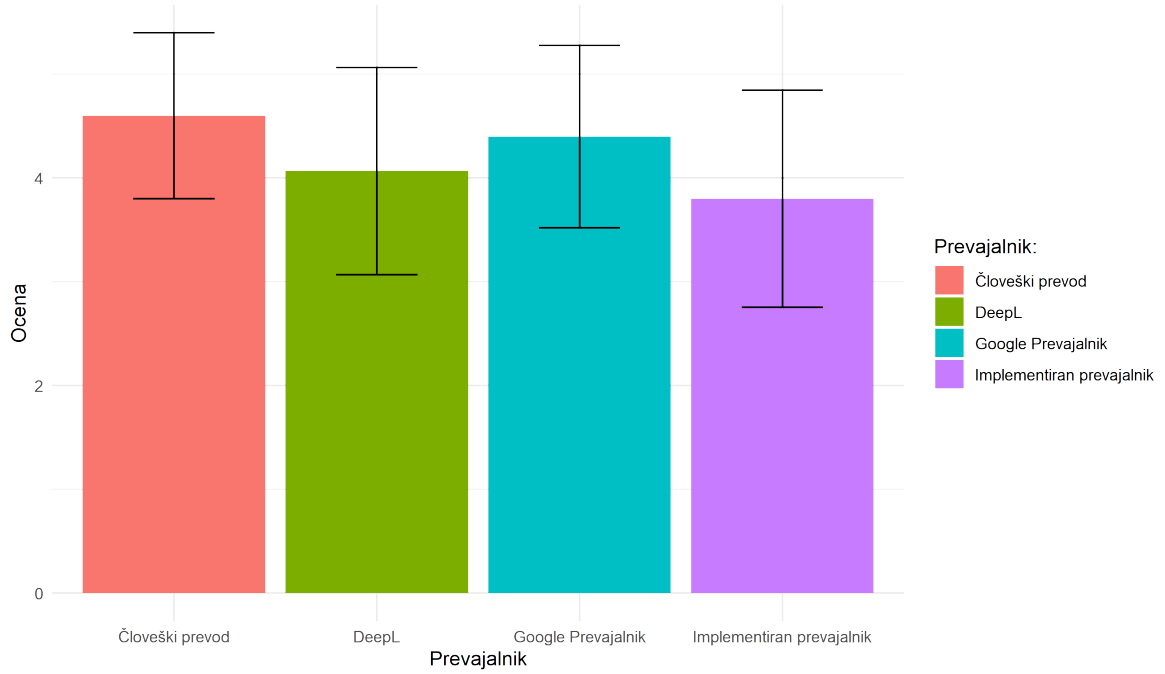
Slika 9: Graf s povprečnimi ocenami 3. referenčne povedi

Prevode vseh štirih prevajalnikov za referenčno poved številka 4 so anketiranci ocenili približno enako. Iz Tabele 13 se lahko vidi, da je razlika med najslabše ocenjenim (prevajalnik izdelan v sklopu zaključnega dela) in najboljše ocenjenim (človeškim prevodom) prevodom le $0,8$ točke. Podobno se lahko razbere tudi iz modusa, ki je za vse prevode 5.

Tabela 13: Aritmetična sredina, standardni odklon, mediana in modus za 4. referenčno poved

Prevajalnik	Aritmetična sredina	Standardni odklon	Mediana	Modus
Google Prevajalnik	4,400	0,879	5	5
DeepL	4,067	0,998	4	5
Implementiran prevajalnik	3,800	1,046	4	5
Človeški prevod	4,600	0,800	5	5

Do iste ugotovitve se lahko pride tudi s pomočjo grafa, ki je na Sliki 10. Opazi se lahko tudi bistvena razlika v oceni med prevodoma DeepL-ja in Google Prevajalnika. V predhodnih primerih ta razlika ni bila toliko razvidna, kot je za to referenčno poved.



Slika 10: Graf s povprečnimi ocenami 4. referenčne povedi

Razlog za tako usklajene ocene bi lahko bil v medsebojni podobnosti prevodov. Prevodi se paroma med seboj razlikujejo v eni besedi (dveh, če predlog »v« štejemo ločeno) ter v vrstnem redu besed. Razlog za nižje ocene prevodov prevajalnikov od človeškega prevoda so: napačen/neobičajen vrstni red besed v povedi (prevod DeepL) ter napačna oblika besed (»V pomladi« namesto »spomladi« in »zacveti« namesto »cveti«).

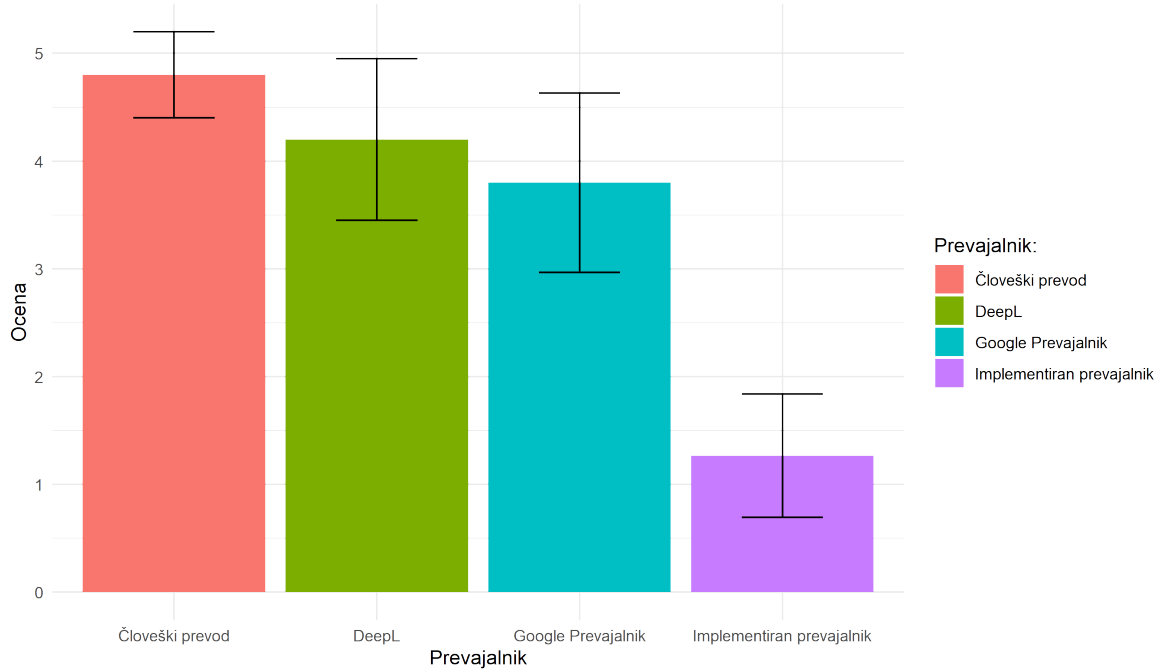
Prevodi, ki so jih izdelali prevajalniki, referenčne povedi številka 5, so bili najslabše ocenjeni. To se lahko opazi iz Tabele 14. Le človeški prevod ima mediano in modus 5, kjer imata za druge prevode tudi DeepL in Google Prevajalnik podobno oceno kot človeški prevod.

Tabela 14: Aritmetična sredina, standardni odklon, mediana in modus za 5. referenčno poved

Prevajalnik	Aritmetična sredina	Standardni odklon	Mediana	Modus
Google Prevajalnik	3,800	0,833	4	3
DeepL	4,200	0,748	4	4; 5
Implementiran prevajalnik	1,267	0,573	1	1
Človeški prevod	4,800	0,400	5	5

Enako se lahko sklepa tudi iz grafa, ki je prikazan na Sliki 11. Opazi se lahko tudi, da je bil prevod prevajalnika, ki je bil izdelan v sklopu zaključne naloge, v tem primeru ocenjen najslabše od vseh petih prevodov. Intervali na Sliki 11 kažejo, da ocene

prevoda, izdelanega s prevajalnikom iz zaključne naloge, nikoli ne dosegajo najslabše ocene preostalihh prevodov.



Slika 11: Graf s povprečnimi ocenami 5. referenčne povedi

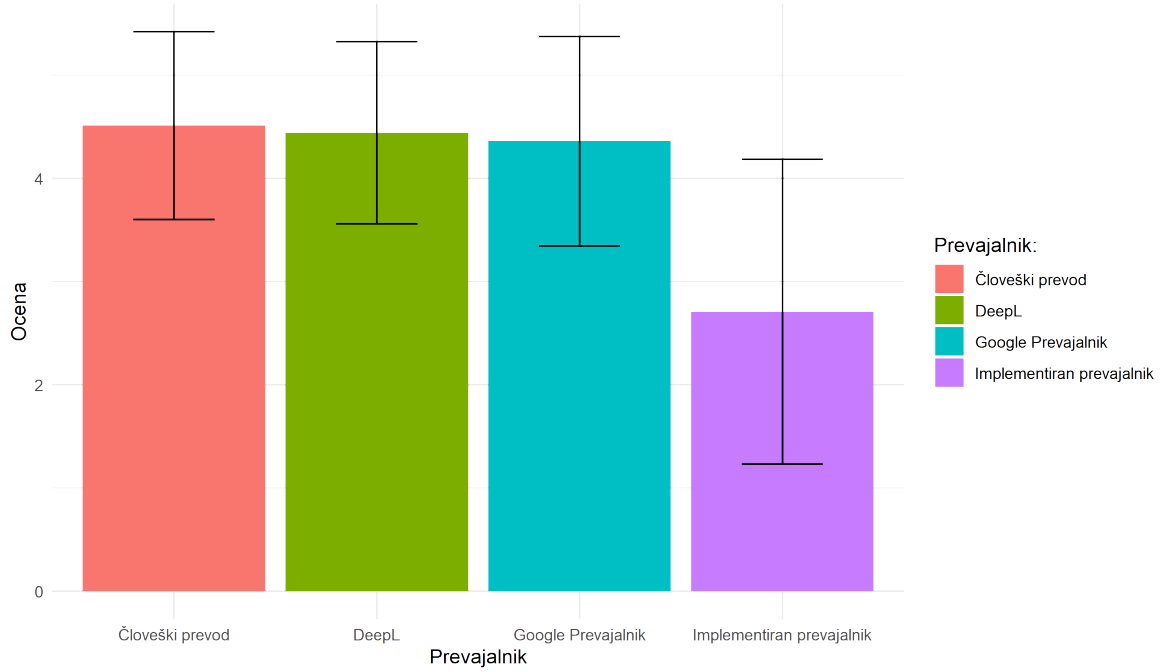
Možen razlog za slabšo oceno prevodov Google Prevajalnika in DeepL je uporaba vikanja namesto tikanja. Čeprav iz italijanske povedi ni razvidno, ali je uporabljeno vikanje ali tikanje, ljudje iz vsebine povedi presodijo, da je bolj primerno tikanje kot vikanje. Nizka ocena prevajalnika, ki je bil izdelan v sklopu zaključne naloge, ni nepričakovana, saj je izdelan prevod brez pomena. Bolj presenetljivo je, da nekateri anketiranci niso ocenili prevoda z oceno 1, kar je razvidno iz standardnega odklona, prikazanega v Tabeli 14 kot tudi na Sliki 11.

Tabela 15: Aritmetična sredina, standardni odklon, mediana in modus za vse refernečne povedi skupaj

Prevajalnik	Aritmetična sredina	Standardni odklon	Mediana	Modus
Google Prevajalnik	4,36	1,015	5	5
DeepL	4,44	0,883	5	5
Implementiran prevajalnik	2,707	1,477	3	1
Človeški prevod	4,51	0,915	5	5

Če se na koncu še izračuna srednje vrednosti in standardni odklon za vse povedi skupaj, se dobi rezultate, ki so predstavljeni v Tabeli 15. Opazi se lahko, da so ocene človeških prevodov in prevodov, izdelanih z Google Prevajalnikom in DeepL medsebojno podobne, bodisi aritmetična sredina bodisi modus in mediana.

Povprečna vrednost in mediana prevajalnika, ki je bil izdelan v zaključni nalogi, je v skladu z opisano interpretacijo rezultata metrike BLEU. Čeprav je modus za prevode, izdelane s tem prevajalnikom 1, zaradi prevoda referenčne povedi 1 in 5, je iz grafa na Sliki 12 razvidno, da mnenje ljudi o tem prevajalniku ni tako nizko, kot ga prikazuje modus.



Slika 12: Graf s povprečnimi ocenami vseh referenčnih povedi skupaj

Možen razlog za slabe ocene prevajalnika, izdelanega v sklopu zaključne naloge, je razviden iz prevodov, ki sta dobila najslabše ocene, in sicer prevoda referenčne povedi številka 1 in 5. V prevodu povedi številka 1 je razvidno nepoznavanje besed, ki jih prevajalnik uporablja pri prevajanju. V prevodih obeh referenčnih povedi pa je vidna izdelava napačnih vzorcev. Oba problema bi se dalo rešiti ali vsaj zmanjšati z uporabo večjih korpusov pri izdelavi modela. Poleg tega še vedno ostane problem specifičnosti prevajalnika, ki je odvisna od vsebine korpusa, uporabljenega za izdelavo prevajalnika.

5 Zaključek

Namen zaključne naloge sta izdelava in evalvacija statističnega strojnega prevajanja iz italijanščine v slovenščino. Pred samo izdelavo prevajalnega sistema so bili predstavljeni matematično ozadje statističnega strojnega prevajanja, obstoječi prevajalni sistemi ter strojno prevajanje in njrgove vrste. Podrobneje sta bila opisana prevajalna sistema DeepL ter Google Prevajalnik, saj sta bila uporabljena kot primerjava s prevajalnikom, ki je bil izdelan v sklopu zaključne naloge.

Pri sami izdelavi prevajalnega sistema je bilo uporabljeno odprtokodno orodje za izdelavo in poganjanje prevajalnih sistemov Moses. Pri tem je v delu opisan celoten postopek izdelave prevajalnega sistema.

Prevajalni sistem, ki je bil izdelan v sklopu zaključne naloge, je bil evalviran s pomočjo metrike BLEU in s človeško oceno prevodov. Ocena BLEU izdelanega prevajalnika je 28,15, kar pomeni, da so prevodi razumljivi, vendar vsebujejo manjše jezikovne napake. Razlog za visoko oceno metrike BLEU je prisotnost besed v izdelanem prevodu, ki so tudi del referenčnega prevoda, četudi izdelani prevod nima pravega pomena.

S človeško oceno prevodov je bila izdelana tudi primerjava prevajalnega sistema z Google Prevajalnikom, prevajalnikom DeepL in človeškim prevodom. Anketiranci menijo, da so prevodi, izdelani z Google Prevajalnikom in DeepL, približno enako kakovostni kot človeški prevodi. Menijo tudi, da prevajalnik, ki je bil izdelan v sklopu zaključne naloge, izdeluje razumljive prevode z večjimi jezikovnimi napakami. Anketiranci so v povprečju ocenili kot razumljive z manjšimi jezikovnimi napakami 60 % prevodov referenčnih povedi, izdelanih s prevajalnikom, ki je bil izdelan v sklopu zaključne naloge, ter 40 % kot nerazumljivih. To je znižalo končno oceno prevajalnika. Razlog za slabo oceno, pridobljeno s človeškim ocenjevanjem, pa je nesoglašanje anketirancev pri prevodih, ki so jih ocenili kot razumljive, ter močno soglašanje pri nerazumljivih prevodih. Do nesoglašanja pride, saj je percepcija jezikovnih napak subjektivna in vezana na rabo ter odnos do jezika, ki ga imajo anketiranci.

Pri tem se je izkazalo, da izdelan prevajalnik ni primeren za prevajanje splošnih besedil, čeprav je polovico referenčnih povedi, katerih vsebina je bolj splošna oziroma vsakdanja, prevedel po mnenju anketirancev dobro. Glede na slaba prevoda referenčnih povedi in dober prevod povedi z vsebino o Evropski uniji lahko prevajalnik

uporabljamo za grobo prevajanje evropskih dokumentov. Kakovost prevodov bi bilo še mogoče povečati z uporabo večjega korpusa, s katerim bi zmanjšali število neznanih besed (OVV) ter izboljšali statistični prevajalni in statistični jezikovni model. Če se želi izdelati prevajalnik, ki je primeren za prevajanje bolj splošne oziroma vsakdanje vsebine, se potrebuje, korpuse z vsakdanjo ali splošno vsebino, zakar bi potrebovali bistveno večje korpuse, kot jih potrebuje za izdelavo prevajalnika, ki je omejen na specifično področje.

6 Literatura in viri

[1] J. Tiedemann, Parallel Data, Tools and Interfaces in OPUS. V *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 2012, 2214–2218.

[2] J. Vičič, Strojno prevajanje in slovenščina. V *Zbornik Sedme konference JEZIKOVNE TEHNOLOGIJE*, 2010, 47–52.

[3] P. Koehn, F. Och in D. Marcu, Statistical Phrase-Based Translation. V *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 2003, 127–133.

[4] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra in R. L. Mercer, The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* 19 (1993) 263–311.

[5] K. Papineni, S. Roukos, T. Ward in W.J. Zhu, Bleu: a Method for Automatic Evaluation of Machine Translation. V *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002, 311–318.

[6] *Moses statistical machine translation system*, `https://www.statmt.org/moses/`. (Datum ogleda: 15. 4. 2021.)

[7] K. Heafield, KenLM: Faster and Smaller Language Model Queries. V *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 2011, 187–197.

[8] F. J. Och in H. Ney, Improved Statistical Alignment Models. V *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, 2010, 440–447.

[9] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, J. D. Lafferty, R. L. Mercer in P. S. Roossin, A Statistical Approach to Machine Translation. *Computational Linguistics* 16 (1990) 79–85.

[10] A. Lopez, Statistical Machine Translation. *ACM Computing Surveys* 40 (2008) 1–49.

[11] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin in E. Herbst, Moses: Open Source Toolkit for Statistical Machine Translation. V *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, 2007, 177–180.

[12] A. Lavie, Evaluating the Output of Machine Translation Systems. V *Proceedings of Machine Translation Summit XIII: Tutorial Abstracts*, 2011, 1–86.

[13] *How does DeepL work?*, DeepL. `https://www.deepl.com/en/blog/how-does-deepl-work`. (Datum ogleda: 15. 4. 2022.)

[14] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes in J. Dean, Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *ArXiv* (2016) 1–23.

[15] *A Neural Network for Machine Translation, at Production Scale*, Google. `https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html`. (Datum ogleda: 15. 4. 2022.)

[16] N. Reimers in I. Gurevych, Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. V *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020, 4512–4525.

[17] R. Skadins, J. Tiedemann, R. Rozis in D. Deksne, Billions of Parallel Words for Free: Building and Using the EU Bookshop Corpus. V *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014, 1850–1855.

[18] R. Steinberger, M. Ebrahim, A. Poulis, M. Carrasco-Benitez, P. Schluter, M. Przybyszewski in S. Gilbro, An overview of the European Union's highly multilingual parallel corpora. *Language Resources and Evaluation* 48 (2014) 679–707.

[19] P. Koehn, *Statistical Machine Translation*, Cambridge University Press, Cambridge, 2009.

[20] L. Adkins in R. Adkins, *The keys of Egypt : the race to read the hieroglyphs*. HarperCollins, 2000.

[21] *Bing Microsoft Translator*, `https://www.bing.com/translator`. (Datum ogleda: 7. 6. 2022.)

[22] *Watson Language Translator IBM*, `https://www.ibm.com/cloud/watson-language-translator`. (Datum ogleda: 7. 6. 2022.)

[23] *Amazon Translate – Neural Machine Translation - AWS*, `https://aws.amazon.com/translate/`. (Datum ogleda: 7. 6. 2022.)

[24] *Amebis Presis*, `https://presis.amebis.si/`. (Datum ogleda: 7. 6. 2022.)

[25] *Prevajalnik Spletni-slovar.com*, `https://www.spletni-slovar.com/prevajalnik`. (Datum ogleda: 7. 6. 2022.)

[26] D. Bahdanau, K. Cho in Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate. *ArXiv* (2014) 1–15.

Priloge

A Anketni vprašalnik

Spoštovani!
Moje ime je Jani Suban in sem študent dodiplomskega študija Računalništva in informatike na Fakulteti za matematiko, naravoslovje in informacijske tehnologije. V okviru zaključne naloge na temo statističnega strojnega prevajanja iz italijanščine v slovenščino želim izvedeti mnenje ljudi, ki govorijo oba jezika, o kakovosti prevodov. Pri tem je podana lestvica, s pomočjo katere boste ocenili prevode. Prevodi so bili izdelani z Google Prevajalnikom, DeepL in prevajalnikom, ki je bil izdelan v sklopu zaključne naloge. Poleg prevodov, izdelanih s strojnim prevajanjem, so podani še prevodi, ki jih je izdelal človek.
Anonimnost anketirancev je zagotovljena. Podatki se bodo uporabljali izključno za potrebe zaključne naloge.
Vaše sodelovanje v raziskavi je prostovoljno. Za reševanje ankete boste potrebovali približno 5 minut.
Če imate kakršno koli vprašanje, me lahko kontaktirate na: 89181056@student.upr.si

Q1 -
Podano je besedilo v Italiajanščini: **"Il tempo è bello, usciamo oggi pomeriggio?"**
S pomočjo spodnje lestvice ocenite prevode, ki so podani v spodnji tabeli.
Lestvica za ocenjevanje prevodov:

1. Poved je nejasna in brez pomena.
2. Jezik poved je nejasen, a se vseeno da razumeti njen pomen.
3. Prevod je razumljiv z večjimi slovničnimi napakami.
4. Prevod je razumljiv z manjšimi slovničnimi napakami.
5. Prevod je razumljiv z minimalnimi slovničnimi napakami.

	1	2	3	4	5
Vreme je lepo, gremo popoldne ven?	○	○	○	○	○
Vreme je lepo, gremo popoldne ven?	○	○	○	○	○
Evropski čas je čudovito, usciamo danes popoldne?	○	○	○	○	○
Gremo danes popoldne ven, saj je lepo vreme?	○	○	○	○	○

Q2 -

Podano je besedilo v Italiajanščini: **"La Commissione Europea delibera sulle nuove leggi."**

S pomočjo spodnje lestvice ocenite prevode, ki so podani v spodnji tabeli.

Lestvica za ocenjevanje prevodov:

1. Poved je nejasna in brez pomena.
2. Jezik poved je nejasen, a se vseeno da razumeti njen pomen.
3. Prevod je razumljiv z večjimi slovničnimi napakami.
4. Prevod je razumljiv z manjšimi slovničnimi napakami.
5. Prevod je razumljiv z minimalnimi slovničnimi napakami.

	1	2	3	4	5
Evropska komisija razpravlja o novih zakonih.	○	○	○	○	○
Evropska komisija razpravlja o novih zakonih.	○	○	○	○	○
Evropska komisija odloča o novih zakonov.	○	○	○	○	○
Evropska komisija odloča o novih zakonih.	○	○	○	○	○

Q3 -

Podano je besedilo v Italiajanščini: **Novak è il più frequente cognome in Slovenia."**

S pomočjo spodnje lestvice ocenite prevode, ki so podani v spodnji tabeli.

Lestvica za ocenjevanje prevodov:

1. Poved je nejasna in brez pomena.
2. Jezik poved je nejasen, a se vseeno da razumeti njen pomen.
3. Prevod je razumljiv z večjimi slovničnimi napakami.
4. Prevod je razumljiv z manjšimi slovničnimi napakami.
5. Prevod je razumljiv z minimalnimi slovničnimi napakami.

	1	2	3	4	5
Novak je najpogostejši priimek v Sloveniji.	○	○	○	○	○
Novak je najpogostejši priimek v Sloveniji.	○	○	○	○	○
Novak je najpogostejša priimek v Sloveniji.	○	○	○	○	○
Novak je najpogostejši priimek v Sloveniji.	○	○	○	○	○

Q4 -

Podano je besedilo v Italiajanščini: **"In primavera la maggior parte delle piante fiorisce."**

S pomočjo spodnje lestvice ocenite prevode, ki so podani v spodnji tabeli.

Lestvica za ocenjevanje prevodov:

1. Poved je nejasna in brez pomena.
2. Jezik poved je nejasen, a se vseeno da razumeti njen pomen.
3. Prevod je razumljiv z večjimi slovničnimi napakami.
4. Prevod je razumljiv z manjšimi slovničnimi napakami.
5. Prevod je razumljiv z minimalnimi slovničnimi napakami.

	1	2	3	4	5
Spomladi večina rastlin zacveti.	○	○	○	○	○
Spomladi cveti večina rastlin.	○	○	○	○	○
V pomladi večina rastlin cveti.	○	○	○	○	○
Spomladi večina rastlin cveti.	○	○	○	○	○

Q5 -

Podano je besedilo v Italiajanščini: **Čome va con la scrittura della tesi di laurea?"**

S pomočjo spodnje lestvice ocenite prevode, ki so podani v spodnji tabeli.

Lestvica za ocenjevanje prevodov:

1. Poved je nejasna in brez pomena.
2. Jezik poved je nejasen, a se vseeno da razumeti njen pomen.
3. Prevod je razumljiv z večjimi slovničnimi napakami.
4. Prevod je razumljiv z manjšimi slovničnimi napakami.
5. Prevod je razumljiv z minimalnimi slovničnimi napakami.

	1	2	3	4	5
Kako ste s pisanjem diplomske naloge?	○	○	○	○	○
Kako poteka pisanje vaše diplomske naloge?	○	○	○	○	○
Kot je treba s znanja in trditev diplomo?	○	○	○	○	○
Kako gre pisanje diplomske naloge?	○	○	○	○	○

Q6 - Spol

- ○ Moški
- ○ Ženska

Q7 - Starost

- ○ Do 14 let
- ○ Med 15 in 19 let
- ○ Med 20 in 24 let
- ○ Med 25 in 29 let
- ○ Med 30 in 34 let
- ○ Med 35 in 39 let
- ○ 40 ali več let

Q8 - V spodnji tabeli označite nivo uporabe slovenščine in italijanščine.

	Materni jezik	Jezik okolja	Tuj jezik
Slovenščina	○	○	○
Italijanščina	○	○	○