Izdelava slovensko-srbskega vzporednega korpusa podnapisov za razvoj strojnega prevajanja v projektu SUMAT

Mirjam Sepesy Maučec, Marko Presker, Danilo Zimšek, Matej Rojc, Damjan Vlaj, Darinka Verdonik, Zdravko Kačič

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko
Smetanova 17, SI-2000 Maribor
mirjam.sepesy@uni-mb.si, marko.presker@uni-mb.si, danilo.zimsek@uni-mb.si, matej.rojc@uni-mb.si, damjan.vlaj@uni-mb.si, darinka.verdonik@uni-mb.si, kacic@uni-mb.si

Povzetek

Prispevek predstavlja izkušnje pri izdelavi vzporednega korpusa podnapisov, namenjenega za učenje modelov strojnega prevajanja. Opisane so značilnosti izvornega gradiva, procesi predpriprave gradiva, ki vključujejo pretvorbe v enoten format in enotno kodiranje, identifikacijo jezika in poravnavanje datotek, tokenizacijo in razcep na povedi, ter postopki poravnavanja ter rezultati evalvacije poravnanih podnapisov in povedi. Vzporedni korpus podnapisov za srbščino in slovenščino je bil razvit v okviru evropskega FP7 projekta SUMAT, katerega cilj je razvoj spletne aplikacije za strojno prevajanje podnapisov.

Building the parallel Slovene-Serbian corpus of subtitles for machine translation in the SUMAT project

The paper describes experiences in building parallel corpus of subtitles, aimed for usage in machine translation. We describe characteristics of the source data, pre-processing (which includes conversion to common format and common coding, language identification and file alignment, tokenization and sentence splitting), subtitle and sentence alignment, and results of alignment evaluation. The parallel corpus of Slovene-Serbian subtitles was developed within the FP7 EU-funded project, named SUMAT, which aims to develop an online service for machine translation of subtitles.

1. Uvod

Podnaslavljanje je priljubljen način za posredovanje tujejezičnih multimedijskih vsebin v veliko evropskih državah in za večino žanrov. Trenutna evropska politika (European Commission, 2010) podpira podnaslavljanje v javnih televizijskih mrežah in posledično se je potreba po podnaslavljanju v avdiovizualni industriji v preteklih letih povečala (MCG, 2007).

Hkrati se podnaslavljanje srečuje s pomembnimi problemi, kot so visoki stroški, časovna potratnost in posledično vprašanje kvalitete podnapisov. Določene raziskave (npr. Volk, 2008; de Sousa et al., 2011) nakazujejo, da bi lahko v prevajanje podnapisov uspešno vključili strojno prevajanje in tako pomagali rešiti navedene probleme.

Trenutno ne obstajajo orodja, ki bi zagotavljala avtomatsko podnaslavljanje gradiv v tujem jeziku. Ena osnovnih ovir je pomanjkanje ustreznih vzporednih korpusov, potrebnih za razvoj modelov za strojno prevajanje. Eden redkih dostopnih korpusov je OPUS OpenSubtitle corpus (Tiedemann, 2009), ki pokriva precej evropskih jezikov, problem pa je, da temelji na odprto dostopnih prevodih s spleta, za katere ni nobenega zagotovila o njihovi kvaliteti.

Po drugi strani so profesionalni prevodi podnapisov večinoma last podjetij, ki se ukvarjajo s podnaslavljanjem in ki praviloma skrbno ščitijo te vire, zato je do njih izredno težko dostopati. Prav tako so formati teh podnapisov zelo različni in nekateri od njih lastniški, npr. Softelovi .o32, .x32 in .s32, Screenov .890, Poliscriptov .pac ali EZTitlesov .ezt.

Z namenom, da se izdela spletno aplikacijo za podnaslavljanje za različne evropske jezike, se je v letu 2011 začel evropski projekt SUMAT[1] (An Online Service for SUbtitling by MAchine Translation), katerega partner je poleg podjetij, ki se ukvarjajo s podnapisi, ter tujih raziskovalnih institutov tudi Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko. V projektu bodo pokriti naslednji jezikovni pari: nemško, francosko, špansko, nizozemsko, švedsko in portugalsko v povezavi z angleščino ter slovensko v povezavi s srbščino[2]. Spletna aplikacija bo komercialni produkt[3], za nekomercialno rabo bo odprta le v okrnjeni funkcionalnosti.

Eden od pomembnih korakov pri izdelavi sistema za strojno prevajanje podnapisov je izdelava vzporednega korpusa podnapisov, potrebnega za učenje prevajalnika. Namen tega prispevka je predstaviti izkušnje pri izdelavi slovensko-srbskega vzporednega korpusa podnapisov v okviru projekta SUMAT. Poravnavanje podnapisov ima v primerjavi s poravnavanjem drugih besedil v vzporedni korpus kar nekaj posebnosti, kot so različni formati izvornih datotek, jezikovne posebnosti prevajanja podnapisov, ki je znano po mnogih redukcijah, preklapljanje med povedmi in podnapisi (podnapis namreč ne sovpada s povedjo) itn.

Članek v nadaljevanju predstavlja značilnosti izvornega gradiva v drugem poglavju, v tretjem poglavju

[1] http://www.sumat-project.eu; projekt financira EU po pogodbi ICT-PSP-270919.

[2] Za slovenski jezik pri prevajalskih podjetjih ni bilo ustrezne količine angleških podnapisov, uporaba para slovenščina-srbščina pa je predvidena predvsem kot posredno prevajanje med pari angleščina-slovenščina in angleščina-srbščina, kadar za posameznega od teh parov že obstaja prevod.

[3] Komercialna narava končnega produkta je med drugim pogojena z vrsto evropskega programa, v okviru katerega je projekt sofinanciran, to je ICT PSP (http://ec.europa.eu/information_society/activities/ict_psp/about/index_en.htm).

potrebne korake predpriprave gradiva ter v četrtem poglavju izkušnje iz poravnavanja in rezultate evalvacije.

2. Izvorno gradivo

Izvorno gradivo so iz svojih arhivov posredovala tri mednarodna podjetja, ki so specializirana za prevajanje podnapisov. Podjetja so zagotavljala, da gre za visoko kvalitetne podnapise, saj je vsak prevod pregledan na več nivojih, preden je posredovan naročniku.

Prejšnji eksperimenti (Volk, 2008) so pokazali, da je kvaliteta avtomatskih prevodov močno odvisna od žanra filma. Ločevanje gradiva po žanrih je temeljilo na klasifikaciji gradiva pri gradivodajalcih, tako da so že ti posredovali datoteke ločene po posameznih žanrih. V osnovi so bili žanri razdeljeni na tiste, ki izhajajo iz vnaprej napisanih scenarijev (ang. scripted), kot so na primer dokumentarni filmi, serije, novice, in tiste, ki predstavljajo spontano govorjene žanre (ang. unscripted), na primer pogovorne oddaje, intervjuji itd. Skupaj so bili žanri razdeljeni na 22 domen.

Poleg datotek s prevodi smo zbirali tudi samo enojezične datoteke, saj je pomembna komponenta prevajalnika tudi jezikovni model.

Posredovanje datotek je potekalo prek FTP-strežnika.

Za jezikovni par slovenščina-srbščina je bilo zbranih 825 datotek, ki so obsegale skupno 169.654 podnapisov v slovenščini in 219.139 podnapisov v srbščini. Datoteke so pripadale naslednjim žanrom: novice, serije in dokumentarni filmi. Več kot pol datotek za par slovenščina-srbščina ni imelo opredeljenega žanra in so bile kategorizirane kot drugo.

Že začetna statistična analiza je pokazala, da imajo v povprečju srbske datoteke več ponapisov kot slovenske. Pozneje se je tudi pokazalo, da datoteke slovensko-srbskega jezikovnega para niso nastale z navzkrižnim prevajanjem, ampak sta slovenski in srbski prevod nastala neodvisno drug od drugega, in sicer na podlagi avdio datoteke (ne na podlagi pisnega scenarija!) v angleščini. To dejstvo je prineslo številne težave v nadaljnji obdelavi gradiva, tako da kljub sicer jezikovno brezhibnemu materialu zbrano gradivo ni zagotavljalo želenih lastnosti za učenje strojnega prevajanja. Tudi količina zbranega gradiva za slovenščino in srbščino je bila manjša kot za druge jezikovne pare.

Televizijski podnaslovi pa so tudi sicer jezikovno nekoliko posebni. Podnaslavljanje je namreč zaznamovano z omejitvami prostora (največ dve vrstici) in časa, zapisovanjem posebnosti govora, prisotnostjo slike in s tem, da pri prevajanju ni nujno osnova izvorni tekst (scenarij), ampak je lahko to tudi zvočni/video posnetek. Zato je pri podnaslavljanju, kot pravi Kovačič (1996: 298), vprašanje ne samo, kako prevesti, ampak najprej kaj prevesti in kaj izpustiti. Redukcije, zgoščevanja, parafraziranje, tako na besedni kot na stavčni ravni, vse to je zelo pogost element podnaslavljanja. V gradivu projekta SUMAT so zato zelo pogosti prevodni pari takšni:

SL: *Naslednjega kar testirajte.*
SR: *Sledeći put kada budem našla dečka, odvešću ga tamo.*

ali:

SL: *Izjavljam, da se bom upokojil.*
SR: *Sada zvanično podnosim ostavku!*

Vse te značilnosti predstavljajo težavo pri poravnavanju in tudi pri nadaljnji uporabi gradiva za učenje modelov strojnega prevajanja.

3. Predpriprava gradiva

Predpriprave izvornega gradiva vključujejo naslednje korake: pretvorbe v enoten format in enotno kodiranje znakov, identifikacijo jezika v datotekah, poravnavanje datotek, tokenizacijo in razcep po povedih.

3.1. Pretvorbe v enoten format

Za učenje prevajanja je v projektu SUMAT uporabljeno orodje MOSES (Koehn idr., 2007), ki zahteva vhodne datoteke v formatu txt. To je bil posledično ciljni format datotek tudi v korpusih SUMAT.

Izvorne datoteke so bile pretežno v formatu pac in različnih izpeljankah formata txt, zato je bil razvit namenski program, ki je vse datoteke pretvoril v enoten txt-format. Največ težav pri pretvorbi je bilo z datotekami v formatu pac, ki je lastniški.

Za kodiranje znakov je bil sprejet dogovor, da bo enoten format UTF-8. Partnerji v projektu so izdelali namenski program za detekcijo in pretvorbo kodiranja. V nekaterih slovensko-srbskih datotekah smo kljub temu zaznali, da imajo črke c, č in ć ter d in đ enake kode. Te datoteke so bile iz korpusa izločene, saj popravljanje te napake ni izvedljivo na preprost način.

Po teh pretvorbah je bila vsebina datotek predstavljena na način, kot prikazuje slika 1. Vsak podnapis ima zaporedno številko, sledita mu časovni kodi za začetek in konec. Besedilo podnapisa je zapisano v eni ali dveh vrsticah.

```
0003    00:00:18:05    00:00:25:15
Ne vem . Ne trdim ,
da vse vem ali da se dobro poznam .

0004    00:00:25:21    00:00:29:21
Vsak dan se spoznavam . O nekom
ne moreš imeti napačne predstave .
```

Slika 1: Izsek iz datoteke v poenotenem formatu

3.2. Identifikacija jezika in poravnavanje datotek

Prenos datotek iz arhivov podjetij na projektni strežnik je bil izveden ročno. To pomeni, da napake pri prenosu niso izključene. Čeprav je del imena datoteke tudi koda jezika, smo jezik v dokumentu identificirali s programom Lingua:Ident (http://search.cpan.org/\~mpiotr/Lingua-Ident-1.6/Ident.pm), ki temelji na verjetnostnem algoritmu na osnovi trigramov črk. Program je bil učen na korpusu OpenSubtitle v.2 za srbščino (Tiedemann, 2009) in Europarl v.6 za slovenščino (Koehn, 2005). S pomočjo omenjenega programa smo iz korpusa uspešno izločili dve napačni datoteki.

Če so datoteke s prevodi generirane iz iste predloge, je upravičeno pričakovati, da se časovne kode v datotekah ujemajo, zato smo v prvem koraku razvili program za poravnavanje datotek na osnovi podobnosti časovnih kod. Program temelji na dinamičnem programiranju in pri primerjavi časovnih kod upošteva vnaprej definirano odstopanje. V projektu je bil sprejet dogovor, da je

dovoljeno odstopanje do 1 sekunde. Vendar program pri slovensko-srbskem jezikovnem paru ni bil uspešen, saj je zaznal le 8 parov datotek od 380. Razlog je bil, da so slovenski in srbski prevodi v SUMAT-ovem gradivu nastajali neodvisno drug od drugega in se časovne kode niso ujemale.

Poravnavanje datotek smo tako izvedli na osnovi imen datotek. Analiza imen je namreč pokazala, da imajo datoteke, ki so pari, v delu imena enako kodo. Tabela 1 kaže primer takih parov datotek.

Datoteka s slovenskim prevodom	**Datoteka s pripadajočim srbskim prevodom**
Glamour's 50 Biggest Fashion Do's and Don'ts_101_Glamour's 50 Biggest Fashion Do's and Don'ts_GBFD0101A_SLV.PAC	GBFD_101_Glamour's 50 Biggest Fashion Do's and Don'ts_GBFD0101A_SRP.PAC
TOO YOUNG TO KILL 15 SHOCKING CRIMES_TOO YOUNG TO KILL 15 SHOCKING CRIMES_101_TYKA0101A-SLV_NEW.pac	TYK_Too Young to Kill - 15 Shocking Crimes_TYKA0101A-SRP_NEW.PAC

Tabela 1: Poravnavanje datotek na osnovi imen datotek

Datoteke, pri katerih v imenu nismo avtomatsko zaznali skupnega niza znakov, je bilo treba poravnati ročno. Tako smo dobili na koncu 380 parov datotek.

3.3. Tokenizacija in razcep po povedih

Prvi korak na vhodnem tekstu predstavlja tokenizacija. Vhod v modulu za tokenizacijo je besedilo v standardu UTF-8.

Vse pomenske enote besedila smo opisali z uporabo regularnih izrazov, tudi morebitne okrajšave in akronime. Detekcija okrajšav in akronimov je podprta z naborom okrajšav in akronimov, predstavljenim v obliki končnega stroja (Finite State Machine – FSM), ki je bil sestavljen na osnovi korpusa FidaPLUS (www.fidaplus.net).

Glavni del tokenizatorja je končni stroj, ki smo ga uporabili v vlogi procesa tokenizacije (prim. Rojc, 2007). Osnovno abecedo smo najprej razširili z znaki UTF-8, npr.: alphabet1 [A-Za-z\\0-\x7F\\xC2-\xDF\\x80-\xBF\^]. Modul tokenizacije je izveden v obliki povezanega seznama (linked list – dequeue), saj mora povezovati in obdelovati več virov informacij med procesom tokenizacije. Tudi na tem nivoju je bilo treba poskrbeti za podporo procesiranju nizov UTF-8.

Proces tokenizacije se izvaja znotraj zanke, vse dokler imamo na vhodu besedilo. Tokeni se med procesiranjem shranjujejo v povezani seznam, dokler ne zaznamo konec povedi. V okviru projekta smo predvideli naslednje tipe tokenov: ločila, besede, akronime, glavne številke, vrstilne številke, decimalna števila itd. Normalizacija tokenov v projektu ni bila potrebna, zato smo jo onemogočili.

```
0       00:00:57,007    00:00:59,924
Peter, general prihaja.

1       00:01:00,051    00:01:02,421
Kako gre? -Ne preveč dobro.
```

```
0       00:00:56,484    00:00:59,444
Piter, general dolazi.

1       00:00:59,524    00:01:01,964
Kako ide sada? -Ne baš dobro, gospodine.
```

(a)

```
0 00:00:57:00 00:00:59:23
Peter , general prihaja .

1 00:01:00:01 00:01:02:10
Kako gre ? -Ne preveč dobro .
```

```
0 00:00:56:12 00:00:59:11
Piter , general dolazi .

1 00:00:59:13 00:01:01:24
Kako ide sada ? -Ne baš dobro , gospodine .
```

(b)

```
peter , general prihaja .
kako gre ?
-ne preveč dobro .
```

```
piter , general dolazi .
kako ide sada ?
-ne baš dobro , gospodine .
```

(c)

Tabela 2: Vhod za tokenizacijo (a), rezultat po tokenizaciji (b) in razcep na povedi (c)

Smo pa v mehanizem tokenizacije vključili posebna tokena za označevanje konca povedi in konca vhodnega besedila: EOS in EOF. Token EOS predstavlja samo možen konec povedi, za končno potrditev konca se v naslednjem koraku preveri tudi širši kontekst. Po procesiranju vsake povedi vhodnega besedila se preverja tudi, ali je že nastopil token za konec vhodnega besedila. Če še ni, se nadaljuje proces tokenizacije na naslednji povedi. Predstavljeni mehanizem tokenizacije in razcep po povedih se da preprosto izvesti z uporabo povezanega seznama in pripadajočih funkcij: *gettoken()*, *pushtoken()* in *Fill*(). Funkcija *Fill()* tako uporablja končni stroj in kopiči tokene v povezanem seznamu, jih analizira, opazuje konktest itd., kar je v veliko pomoč v procesu razcepa na povedi. Prvi korak določanja konca povedi izvaja sam končni stroj (označi možen token EOS). Na nivoju povezanega seznama pa nato ob upoštevanju desnega in levega konteksta podamo končno odločitev. Za končni stroj so možni nastopi konca povedi (EOS) na danem vhodnem besedilu: ločila (.!?...), kombinacija znakov \n\n, če se hkrati naslednji token začne z veliko črko ter če token z ločilom (.) ni okrajšava ali akronim. Problem v tem primeru predstavljajo npr. lastna imena, kar smo reševali z obsežno zbirko lastnih imen (Onomastica, slovar LC-STAR (dostopna tudi prek ELDE)).

Vhodne datoteke so vključevale tudi specifične tokene, ki jih je bilo treba ustrezno detektirati in procesirati ter prikazati na izhodu tokenizacije, zlasti številčenja segmentov in časovne kode. Tako smo morali generirati več formatov izhodov. Za en del korpusa je bilo treba to informacijo izločiti, za drugi del korpusa pa ohraniti in ustrezno dodati označenim povedim. Tudi te specifične tokene smo opisali z uporabo regularnih izrazov. Primer vhoda/izhoda modula za tokenizacijo nazorneje prikazujemo v tabeli 2.

Procesu tokenizacije je sledila še pretvorba v male črke.

4. Poravnavanje

Na voljo so različna orodja za avtomatsko poravnavanje besedil po povedih. Gale-Churchev poravnalnik (1993) temelji na verjetnosti, izračunani za vsak par povedi glede na dolžino povedi (število znakov). Uporabljen je bil npr. za poravnavanje korpusov Europarl (Koehn, 2005) in JRC-Acquis (p://langtech.jrc.it/JRC-Acquis.html). Moorov (2002) dvojezični poravnalnik kombinira dolžino povedi in število besed in je uspešen predvsem pri čim daljših korpusih. Hunalign (Varga et al., 2005) kombinira dolžino povedi in slovar, če ni slovarja, pa ga nadomesti z verjetnostmi, izračunanimi na podlagi korpusa. Deluje samo na korpusih z do 20.000 povedmi, daljše korpuse pa razbije na manjše dele. Gargantua (Braune, Fraser, 2010) podobno kot Moorov poravnalnik temelji na podobnosti povedi, razlike so v iskalnih strategijah in klestenju iskalnega prostora. Bleualign (Sennrich, Volk, 2010) uporablja strojni prevajalnik izvornega teksta, zato je manj primeren za uporabo, če tega ni na voljo. Podrobneje primerjajo navedena orodja Abdul-Rauf idr. (2010) in ugotavljajo, da se najbolje obnesejo Bleualign, Gargantua in Hunalign. Ker je za prvega potreben strojni prevajalnik, za drugega pa večji korpus, smo se v projektu SUMAT odločili za poravnalnik Hunalign.

Pri poravnavanju korpusa podnapisov imamo na izbiro dve osnovni enoti poravnavanja: podnapis ali poved. Iz teorije statističnega prevajanja vemo, da so za algoritem učenja primernejše krajše enote. V splošnem so podnapisi krajši od povedi, kakršne so sicer značilne za pisna besedila. Toda če podnapise združimo in razcepimo po povedih, ni nujno tako, saj so posamezne povedi v podnapisih pogosto samo eno- ali dvobesedne fraze, ki so značilne za govorno komunikacijo. Analiza (glej tabelo 3) učnega korpusa SUMAT je pokazala, da so povedi v njem v povprečju krajše od podnapisov. Prav tako so v povprečju povedi/podnapisi v srbskem delu korpusa daljši.

	Slovenščina	Srbščina
dolžina povedi	6,5	6,8
dolžina podnapisa	8,2	8,5

Tabela 3: Povprečne dolžine povedi in podnapisov v učnem korpusu SUMAT

Korpus smo ločeno poravnali po povedih in po podnapisih, da bomo lahko v nadaljevanju primerjali uspešnosti prevajalnih sistemov na osnovi povedi in podnapisov.

4.1. Poravnavanje na osnovi besedila in na osnovi časovnih kod

Za poravnavanje povedi in podnapisov smo najprej uporabili pristop poravnave na osnovi besedila in besedilnih značilnosti s pomočjo orodja Hunalign (Varga et al., 2005). Poravnavanje poteka v več zaporednih iteracijah. Orodje tvori matrike poravnav in izračuna uteži za te poravnave. Te uteži temeljijo na podobnosti dolžine enote in morebitni prisotnosti različnih besed v slovarju. Orodje lahko, če slovarja ne vključimo v postopek poravnave, samo generira slovar, ki ga uporabi v kasnejših iteracijah. V našem primeru se je pokazalo, da generirani slovar ni preveč uporaben, vsako vključevanje lastnih slovarjev pa je rezultat poravnave le še poslabšalo. Primer poravnave na podlagi besedila je prikazan v tabeli 4.

Drugi pristop temelji na podlagi poravnavanja časovnih kod podnapisov. Pri poravnavanju povedi smo sami tvorili časovne kode začetka in konca povedi na podlagi števila besed v povedi. Pri poravnavanju podnapisov, in posledično tudi pri poravnavanju povedi, se je pokazalo, da je poravnavanje zelo odvisno od tolerance (tj. odstopanja časovnih kod), ki smo jo nastavili za še dopustno, saj nekatera prevajalska podjetja ne uporabljajo predlog, kar pomeni, da poleg prevoda spreminjajo tudi začetne in končne časovne kode. Premajhna toleranca pomeni premalo poravnanega materiala, prevelika toleranca pa privede do nepravilnosti, saj dopušča, da se kratke povedi ne poravnajo oz. da se po nepotrebnem dodajo poravnavi. Z našo skripto smo lahko zaznali poravnave 1:1 ali 1:N.

4.2. Problemi pri poravnavanju

Pri poravnavanju korpusa smo naleteli na številne težave. Izvirajo predvsem iz tega, da so prevajalci spreminjali časovne kode in razbijali podnapise na različne dele.

Ena od težav, ki se je pogosto pojavljala, je povezana z zelo kratkimi podnapisi, ki so predvsem v srbskem delu korpusa vključeni, v slovenskem delu korpusa pa izpuščeni. Problem je prikazan na primeru v tabeli 4.

Slovenski prevod	Srbski prevod
0020 00:02:01:11 00:02:06:11 Kako se počutiš v središču ? Je naporno ? Kako se spopadaš s tem ?	0029 00:02:01:20 00:02:05:11 Kako se osećaš zbog ovolike pažnje koju dobijaš ? Zar nije ludo ? 0030 00:02:05:14 00:02:06:15 Jeste ! 0031 00:02:06:19 00:02:07:24 Da li se dobro nosiš sa tim ?

Tabela 4: Prevodi – primer kratkih odgovorov

V srbskem delu korpusa vidimo, da 30. podnapis predstavlja odgovor, ki ga v slovenskem delu korpusa ne najdemo. Če opazujemo časovne kode, pa vidimo, da bi 20. slovenski podnapis poravnali z 29. in 30. srbskim podnapisom, 31. pa bi ostal neporavnan.

Nadaljnja težava je bila, da so povedi v obeh prevodih različno dolge. Slovenski prevodi imajo v povprečju krajše povedi kot srbski prevodi, zato je tudi časovni interval temu primerno krajši. Na primeru v tabeli 5 je prikazano, kaj to pomeni za poravnavanje.

Slovenski prevod	Srbski prevod
00:20:56:15 00:20:59:06 Je to človek ?	00:20:57:04 00:21:01:18 Je li to čovek tamo ili nešto slično ?

Tabela 5: Prevodi – primer različno dolgih povedi

Da bi povedi poravnali, bi potrebovali toleranco vsaj 2 sekund in 12 okvirov, kar pomeni skupaj 2,48 sekunde. Tako velika toleranca pa ni več smiselna, saj je predolga, nekatere povedi so celo krajše od tega.

Poleg navedenih težav so se pojavljale še druge, na primer neoznačena menjava govorca, manjkajoči sklop podnapisov, časovno zamaknjene datoteke ali časovno raztegnjene datoteke ipd.

4.3. Postopek evalvacije

Za evalvacijo poravnavanja je bil iz celotnega korpusa izločen testni nabor, in sicer za vsak jezikovni par 1.000 podnapisov in pripadajočih povedi. Material je bil izbran tako, da je odražal delež zastopanosti posameznih žanrov iz celotnega korpusa, in sicer:

- za evalvacijo poravnavanja povedi smo uporabili 50 vzporednih slovensko-srbskih zaporednih podnapisov, razcepljenih po povedih, iz 10 različnih parov datotek,
- za evalvacijo poravnavanja podnapisov smo uporabili 50 vzporednih slovensko-srbskih zaporednih podnapisov iz 10 različnih parov datotek.

Ročno poravnavanje testnega nabora po povedih in po podnapisih je izvajalo 6 oseb. Navodilo je bilo, da se poravnajo samo tiste povedi in podnapisi, ki jih je mogoče smiselno poravnati; če določen podnapis ali poved v drugem jeziku ni imel para, je ostal neporavnan. Pri ročni poravnavi smo tvorili dva tipa datotek, ene so vsebovale slovenske in druge vzporedne srbske podnapise. Neporavnani podnapisi so bili izločeni. Če je bil istopomenski podnapis v slovenskem jeziku predstavljen v eni vrstici, v srbskem jeziku pa v dveh, smo podnapis v srbskem jeziku predstavili v eni vrstici ter dodali tri znake "~~~" med združenima vrsticama. Na ta način smo lahko uporabili skripte, ki smo jih uporabljali za postprocesiranje poravnav, ki smo jih dobili s pomočjo programa Hunalign. Tabela 6 prikazuje primer takšne ročne poravnave.

Slovenski podnapis
Izvorni
Te pokličeva pozneje . Rada vaju imam ! Je to avtobus za žuranje ?
Ročno poravnan
Te pokličeva pozneje . Rada vaju imam ! Je to avtobus za žuranje ?

Srbski podnapis
Izvorni
Zvaćemo te kasnije . Volim vas . Da li je ovo autobus za zabavu ?
Ročno poravnan
Zvaćemo te kasnije . ~~~ Volim vas . Da li je ovo autobus za zabavu ?

Tabela 6: Primer ročne poravnave podnapisov

Za boljši pregled nad poravnavami in lažjo evalvacijo smo ustvarili še tretjo datoteko, v kateri so bile oštevilčene vrstice podnapisov in usklajene glede na to, katere vrstice so dejanski vzporedni pari prevodov. Primer vsebine takšne datoteke je podan v tabeli 7.

Za slovenske podnapise	**Za srbske podnapise**
1	1
2	3

Tabela 7: Označevanje začetkov vrstic vzporednih prevodov

Številke v tabeli 7 pomenijo, da je vrstica 1 v datoteki s srbskim prevodom prevod vrstice 1 v datoteki s slovenskim prevodom, vrstica 2 v datoteki s srbskim prevodom nima prevoda v slovenskem delu korpusa, vrstica 3 v datoteki s srbskim prevodom pa je prevod vrstice 2 v datoteki s slovenskim prevodom.

4.4. Rezultati evalvacije

Po opravljeni avtomatski poravnavi in ročno poravnanem testnem naboru smo ocenili avtomatsko poravnavo testnega nabora. Ocene smo dodelili po naslednjih formulah:

- pravilnost = tp + tn / (tp + tn + fp + fn),
- natančnost = tp / (tp + fp),
- priklic = tp / (tp + fn),

pri čemer predstavlja tp število pravilno poravnanih enot, tn število pravilno neporavnanih enot, fp število napačno poravnanih enot in fn število napačno neporavnanih enot. Po opravljenem poravnavanju smo izvedli še primerjavo uspešnosti različnih načinov poravnavanj. Rezultate evalvacije prikazuje tabela 8.

Enota	**Poravnava**	**Pravilnost**	**Natančnost**	**Priklic**
poved	časovne kode	45 %	81 %	43 %
poved	besedilo	74 %	74 %	99 %
podnapis	časovne kode	47 %	74 %	49 %
podnapis	besedilo	51 %	54 %	78 %

Tabela 8: Uspešnost poravnavanja korpusa SUMAT

Rezultati evalvacije kažejo nizko uspešnost poravnavanja. Razlog je v gradivu. To smo potrdili z eksperimentom, v katerem smo z algoritmom poravnavanja na osnovi časovnih kod poravnavali korpus OpenSubtitle v.2. Poravnave povedi v tem korpusu so zapisane tudi kot dodatna informacija. Oboje smo primerjali z ročno narejenimi poravnavami. Rezultate prikazuje tabela 9. Poravnave, zapisane v korpusu, smo poimenovali OPUS (tj. ime projekta, v katerem je korpus nastal). Iz tabele 9 je razvidno, da daje naš algoritem poravnavanja na osnovi časovnih kod boljše rezultate kot algoritem, uporabljen pri gradnji korpusa OpenSub. S tem je bila potrjena pravilnost metod, implementiranih v našem algoritmu.

Enota	**Poravnava**	**Pravilnost**	**Natančnost**	**Priklic**
poved	časovne kode	93 %	94 %	98 %
poved	OPUS	85 %	87 %	96 %
podnapis	časovne kode	73 %	81 %	85 %

Tabela 9: Uspešnost poravnavanja korpusa OpenSubtitle v.2

V naslednjem koraku smo se odločali, kateri način poravnavanja ohraniti v ciljnem korpusu. Če primerjamo uspešnost obeh načinov poravnavanj na povedih, vidimo, da je le natančnost poravnavanja na osnovi časovnih kod

večja, priklic in pravilnost pa sta bistveno slabša. Odločili smo se, da v primeru korpusa poravnanih povedi ohranimo poravnave, ki jih je tvorilo poravnavanje na osnovi besedila. V primeru poravnavanja podnapisov razlika v pravilnosti obeh postopkov ni tako izrazita, natančnost pa kaže spet v prid poravnavanja na osnovi časovnih kod. V tem primeru smo se odločili, da korpus poravnanih podnapisov sestavimo iz poravnav, ki jih je tvorilo poravnavanje na osnovi časovnih kod.

V tabeli 10 je podana velikost končnega korpusa SUMAT, ki obsega okrog 110.000 poravnanih enot. Primerjava z obsegom izvornega korpusa kaže veliko izgubo materiala (okrog 40 %). Razlog je verjetno v načinu priprave izvornega gradiva, ki smo ga opisali v drugem poglavju.

Enota	Poravnanih enot v korpusu
poved	110.481
podnapis	112.293

Tabela 10: Vzporedni korpus SUMAT

5. Zaključek

V članku smo opisali postopke, ki smo jih izvedli pri pripravi vzporednega korpusa SUMAT. Korpus bomo uporabili za učenje sistema za strojno prevajanje podnapisov. Korpus omogoča primerjavo uspešnosti sistemov za strojno prevajanje na osnovi podnapisov in povedi. Ker je korpus po obsegu precej skromen in ne dosega želenih lastnosti za optimalno učenje strojnega prevajanja, bo potrebno opiranje na druge obstoječe slovensko-srbske vzporedne korpuse, kot je na primer korpus OpenSubtitles. Korpus SUMAT bo v prihodnje tudi avtomatsko oblikoslovno označen, da bomo lahko preizkusili vpliv oblikoslovnih oznak na kvaliteto statističnega strojnega prevajanja. Korpus žal ne bo dostopen za javnost, ampak samo v okviru projektnega konzorcija, saj podjetja, ki so tekstodajalci, izredno zaščitniško skrbijo za svoja gradiva in ne dovolijo, da bi v kakršni koli obliki prišla v javnost.

6. Literatura

Abdul-Rauf, S., Fishel, M., Lambert, P., Noubours, S., Sennrich, R. (2010). Evaluation of Sentence Alignment Systems. Available at: http://lium3.univ-lemans.fr/mtmarathon2010/ProjectFinalPresentation/SentenceAlignment/sentence_alignment.pdf.

Braune, F., Fraseer, A. (2010). Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. V: COLING 2010, Peking, Kitajska.

European Commision (2010). *Audiovisual Media Services Directive* (AVMSD – 2010/13/EU). Official Journal of the European Union, 10 March 2010.

Gale, W.A., Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Comput. Linguist.* 19(1): 75-102.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation, *MT Summit*.

Koehn, P., 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 and E. Herbst. (2007). Moses: open source toolkit for statistical machine translation. In: ACL 2007: proceedings of demo and poster sessions, Prague, Czech Republic, pp. 177–180.

Kovačič., I. (1996). Subtitling strategies: a flexible hierarchy of priorities. In: *Traduzione multimediale per il cinema, la televisione e la scena / Multimediale Übersetzung für Film, Fernsehen und Bühne / Multimedia translation for film, television and the stage: atti del convegno internazionale*. Forlì, 26-28 October 1995 (pp. 297--305).

Media Consulting Group (2007). *Study on dubbing and subtitling needs and practices in the European audiovisual industry*. On behalf of the Information Society and Media Directorate General and the Culture Directorate of the European Commission, November 2007.

Moore, R. E. (2002). Fast and accurate sentence alignment of bilingual corpora. V: AMTA'02 proceedings, London, Velika Britanija (pp. 135—144).

Rojc, M., Kačič, Z. (2007). Time and space-efficient architecture for a corpus-based text-to-speech synthesis system. *Speech commun*. 49(3): 230-249.

Sennrich, R., Volk, M. (2010). MT-based sentence alignment for OCR-generated parallel texts. V: AMTA'10 proceedings, Denver, Kolorado.

de Sousa, S., Aziz, W., Specia, L. (2011). Assesing the Post-Editing Effort for Automatic and Semi-Automatic Translations of DVD Subtitles. *Proceesings of Recent Advances in Natural Language Processing Conference* (RANLP-2011). Hissar, Bulgaria.

Tiedemann, J. (2009). News from OPUS – A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In: N. Nicolov, K. Bontcheva, G. Angelova,, R. Mitkov (eds.): *Recent Advances in Natural Language Processing* (vol. V) (pp. 237--248). Amsterdam, Philadelphia: John Benjamins.

Varga, D., L. Németh, P. Halácsy, A. Kornai, V. Trón, V. Nagy (2005). Parallel corpora for medium density languages. In: *Proceedings of the RANLP 2005* (pp. 590--596).

Volk, M. (2008). The Automatic Translation of Film Subtitles: A Machine Translation Success Story? In: J. Nivre, M. Dahllof, B. Megyesy (eds.): *Resourfceful language Technology: Festschrift in Honor of Anna Sagvall Hein* (vol 7 of Studia Linguistica Upsaliensia, Uppsala, Sweden) (pp. 202—214).