

UNIwersytet IM. ADAMA MICKIEWICZA
UNIVERSITATO ADAM MICKIEWICZ
Wydział Etnolingwistyki – Etnolingvistika Fakultato

Vitor Tagor de Magalhães Monteiro

Aŭtomataj Tradukiloj
Automatyczne narzędzia do tłumaczeń
Automatic Translation Tools

Finlaboraĵo – Praca końcowa – Final thesis
Studdirekto: Postdiplomaj Interlingvistikaj Studoj
Kierunek studiów: Podyplomowe Studia Interlingwistyki
Study program: Postgraduate Interlinguistic Studies

Temgvidanto: Prof. Christopher Gledhill

Poznań 2024

Antaŭparolo

Ĉi tiu finlaboraĵo staras kiel atesto pri la kompletigo de vojo plena de lernaj, kreskaj kaj transformiĝaj spertoj.

Estas kun profunda dankemo ke mi agnoskas la kontribuojn de miaj instruistoj, kies konstrua kritiko kaj gvidado estis esencaj por identigi mankojn kaj reorientigi mian lernadon.

Kvankam ĝi ne estas mia specialiĝo mi dankas prof. dr-ino Katalin Kováts pro ŝia konstanta sindediĉo al la Instruista Trejnado kaj subteno por la evoluado de ĉiuj studentoj.

Mi estas vere dankema al prof. dr-ino Ilona Koutny pro la kreado de la Interlingvistikaj Studoj kaj kunordigado de tiel ampleksa kampo de studoj.

Speciale mi dankas al prof Christopher Gledhill pro la gvidado en la evoluo de mia verko por ke ĝi estu pli klara kaj povu enfokusigi la pli gravajn temojn.

Mia elkora danko iras al ĉiuj, precipe al mia kara Elazir, al miaj gefiloj Artur kaj Débora, kaj al mia unujara nepo Cauê.

Vitor

Enhavtabelo

Enkonduko.....	6
1 Ŝanĝoj en la tradicia tradukmodelo	9
1.1 Kadro de kompetenteco por tradukado.....	9
1.2 Homa engaĝiĝo en maŝintradukado	10
1.3 Kvalito de la tradukado	12
2 Taŭga formato por la dokumentoj	15
2.1 Formatoj de dosieroj por komputilaj tradukprogramoj.....	15
2.2 Kontrolo de historiaj ŝanĝoj	19
3 Maŝina Tradukado.....	23
3.1 DLT Distribuita Lingvo-Tradukado	24
3.2 SMT Lingvomodeloj.....	25
3.3 Lingvaj modeloj kiel nigra skatolo kaj neŭronaj retoj	26
3.4 NMT Lingvomodeloj	27
3.5 Uzo de la NMT-modeloj.....	29
4 Kodiĝo de la datenoj.....	31
4.1 La Unikoda kodaro.....	31
4.2 Normaligo de la teksto	34
4.3 Ĵetonigo de la teksto	35
5 Uzoj de Neŭronaj Maŝinaj Tradukmodeloj.....	39
6 Kiel malaltigi la tradukkostojn.....	41
6.1 Loka maŝina tradukado	41
6.2 Interreta platformo	41
6.3 Konkludoj pri la traduk kostoj	42
Bibliografio.....	44
Resumo.....	48
Summary.....	49
Sumário.....	50

Enkonduko

Praktike ĉiuj homoj, pere de komputilo aŭ saĝtelefono, havas aliron al la novaj teknologioj, kaj multaj novaj aplikaĵoj nun ekzistas; tio kaŭzis dramatikajn ŝanĝojn en la maniero, kiel homoj interagis, lernas kaj laboras. Tial, lerni novajn manierojn plenumi taskojn estas parto de la moderna vivo. Grandaj teknologiaj entreprenoj ankaŭ konstante adaptiĝas al la ŝanĝoj en ka merkato pro la disvolviĝo de la informteknologio. Mi mem travivis tiujn ŝanĝojn dum mia laboro en du malsamaj grandaj entreprenoj.

Unue, dum la 1980-aj jaroj, kiam mi laboris en la komputila industrio ĉe "*Burroughs Corporation*", tradicia kompanio fondita en 1886, mi atestis en 1986 ĝian kunfandiĝon kun "Sperry UNIVAC" por formi la nuntempe multe malpli konatan kompanion *Unisys*¹. La kialo de la kunfandiĝo ne estis pro la kvalito de la propraj komputiloj, sed la apero de multe pli malgrandaj personaj komputiloj je pli malalta prezo kaj la ŝanĝo de fokuso de la aparataro al la programaj aplikaĵoj. La merkato de la grandaj komputilaj kompanioj malkreskis, ĉar ili devis dividi ĝin kun novaj ludantoj kiel *Microsoft* kaj *Apple*, kaj novaj operaciumoj kiel *Unikso* kaj poste *Linukso*.

Due, jam en ĉi tiu jarcento, laborante ĉe Siemens, fondita en 1847 kiel "*Siemens & Halske Telegraph Construction Company*"², mi atestis alian grandan ŝanĝon. Post pluraj internaj disdividoj ĝi fariĝis en 2008, nomata kiel "*Siemens Enterprise Communications*", komuna entrepreno kun la USONA "*Gores Group*"³, sekve ĝi ŝanĝis sian nomon al "*Unify*" en 2013. En 2016 ĝi estis akirita de "*Atos*"⁴ kaj en 2023 ĝi estis vendita al "*Mitel*"⁵, kiu funkcias en la sama telekomunika merkato. Denove, la kialo de la reorganizo ne estis pro la kvalito de la telefonaj sistemoj, sed pro la apero de inteligentaj telefonoj je pli malaltaj prezoj kaj la ŝanĝo de fokuso de simplaj voĉvokoj al multmediaj telefonaj aplikaĵoj. La merkato de la grandaj entreprenoj pri komunikadaj produktoj malkreskis, ĉar ili devis dividi ĝin kun novaj ludantoj, kaj konsekvence la apero de pluraj programaj produktoj, kiel Skype, Zoom, WhatsApp, Google Meet, Microsoft Teams, ktp.

Io simila povas nun okazi en la tradukindustrio, oni scias ke grandaj firmaoj kiel "*Trados*"⁶ fondita en 1984 estis aĉetita en 2005 de "SDL plc"⁷, kaj kunfandiĝis en 2020 kun "RWS plc"⁸ kiu nun fokusas sur specialigitaj servoj. Nuntempe, por ĉiuj tagaj uzoj, rapidaj tradukoj estas provizitaj per iloj kiel "*DeepL Translate*", "*Google Translate*", "*Microsoft Translate*", ktp, kiuj povas esti uzataj en saĝtelefonaj aparatoj por traduki konversacion, fotografiaĵon de

¹ <https://www.unisys.com/>

² <https://www.siemens.com/>

³ <https://www.gores.com/>

⁴ <https://atos.net/en/>

⁵ <https://www.mitel.com/>

⁶ <https://www.trados.com/>

⁷ <https://www.sdl.com/>

⁸ <https://www.rws.com/>

tekstopaĝo aŭ tajpitan tekston. Ilia interreta funkciado alportas zorgojn pri la privateco de la datenoj, sed montras ke eblas aliro al tradukiloj, kiuj ne nur provizas maŝintradukadon, sed ankaŭ ofertas aliron al profesiaj lingvistoj, kiel *“SmartCat”*⁹.

Oni konstante serĉas ilojn por pli rapide fari onian laboron, kun malpli da peno, ripeteble kaj samtempe konservante fidindecon. En la fako de tradukado estis konstatite jam delonge (Tytler 1813), ke vortaroj kaj gramatikoj sole ne sufiĉas por fari bonajn tradukojn, necesas ankaŭ multe da legado kaj kritika atento. Efektive, Memoroj de Tradukado, Terminaroj kaj korpusoj estas utilaj iloj, sed por kapti la nuancojn de signifo, idiomaĵajn uzojn kaj konstruojn, necesas koni la historian kuntekston, kompreni la temon kaj tipan uzadon de vortoj en la fontolingvo kaj cellingvo. Laŭ tiu ĉi rezonado, povas ŝajni neebla por maŝino fari bonan tradukadon, tamen multaj provoj estis faritaj por uzi maŝinojn por helpi en tradukaj taskoj, pruvante ke ili povas provizi suface bonajn rezultojn por specifaj uzoj.

En tiu ĉi studo la atento estas limigita al tradukado de skribita aŭ parolata lingvo, post kiam ĝi estas konvertita al teksto uzebla per komputilo, kvankam aliaj eblecoj, kiel ekzemple tradukado de teksto al piktogramoj, povas esti utilaj en certaj specifaj uzoj, tamen, tiu ĉi kampo de studo ankoraŭ havas grandajn limigojn. En studo por taksu la eblan uzon de traduko de teksto al piktogramo (Bulté 2021) en flandra migrada kunteksto, oni konkludis, ke la sistemo ne devus esti uzata de migrantoj tiel, kiel ĝi estas. Ĝi povus esti uzata por akiri esencajn tradukojn, sed tio estus ĉefe utila por simplaj, preferinde mallongaj frazoj, kiuj ne enhavas specifajn kaj maloftajn vortojn. La sistemo ankaŭ povus esti utila ĝenerale por faciligi komunikadon kun migrantoj, en situacioj kie la generitaj tradukoj estas kontrolataj.

Sono, bildo kaj video, kiuj iam estis konservitaj en analoga formato, kiel vinilaj diskoj, kasedobendoj, filmobendoj, iom post iom transiris al formato, kiun oni povas prilabori per komputiloj. Hodiaŭ, informo kutime estas konservita en cifereca formato; presitaj libroj unue havas ciferecan kontraŭparton, kiu povas esti montrata en diversaj formatoj, kiel PDF-dosiero, retpaĝo aŭ elektronika libro, ktp. Ankaŭ, pluraj mediotipoj – teksto, sono, bildo kaj video - povas aperi en la sama dokumento. Nuntempe pluraj historiaj dokumentoj estas konvertitaj al cifereca formato, ekzemple, la Projekto Gutenberg¹⁰ disponigas multajn librojn redaktitajn en kontrolita maniero. Aliflanke, optika signorekono estas havebla kiel aplikaĵo eĉ en saĝtelefonoj kaj povas esti libere uzataj. Ĉi tiu simpla konvertiĝo de formatoj, en la sama lingvo, bone funkcias, kaj maŝinoj povas solvi la teknikajn problemojn, tamen traduki tiujn informojn al alia lingvo prezentas grandan defion, ĉar signifas konverti la stilon, senton, ideon kaj signifon al alia kulturo.

Alia problemo estas apartigi la tradukadon de la fina prezentado de la tradukita teksto. Por ebligi ke ekde ununura dokumento, en la fontolingvo, la informo estu prezentata en TTT-paĝo (Tut-Tera Teksaĵo), PDF (Portebla DokumentFormo), EPUB (Elektronika PUBLikigo), ktp, en pluraj cellingvoj, multaj organizaĵoj kolektas informojn kaj eldonas specifojn,

⁹ <https://www.smartcat.com>

¹⁰ <https://www.gutenberg.org/>

datenbankojn kaj procedojn por helpi eldonistojn kaj tradukistojn. Vasta gamo da dosierformatoj, kodaroj, lokaĵaroj, terminaroj, korpusoj, vortaroj, datenbankoj, lingvomodeloj, ktp estas uzata por ebligi la uzon de la samaj dokumentoj, operaciumoj, programoj, filmoj, artefaritaj tradukiloj, ktp en diversaj malsamaj internaciaj kuntekstoj.

Ĉi tiu studo celas fronti la problemon de konstanta ĝisdatigado kaj tradukado de dokumentoj al pluraj lingvoj, kiu okazas, ekzemple, en la entrepreno en kiu mi nun laboras. La pluraj transformiĝoj kaj kunfandado de la organizacio postulis ne nur ŝanĝojn de nomoj de la kompanio kaj produktoj, sed ankaŭ reuzon de la sama informoj en aliaj pluraj kuntekstoj. Ekzemple, la telefonia servilo de banko-telefonvokocentro kundividis partojn de dokumentoj kun la telefonia servilo de publika urĝa vokocentro kiel NG911¹¹ en Usono aŭ NG112¹² en Eŭropo, kvankam ili havas sufiĉe malsamajn funkciojn.

La esplorado fokuziĝos en trovi taŭgan formaton, kiu ebligas reuzi sekciojn kaj temojn de la dokumentoj, reuzi antaŭajn tradukojn kaj terminologiajn datenbankojn, kontroli ĝiajn historiajn ŝanĝojn, certigi la kvaliton de la tradukadoj kaj samtempe, kiel eble, malaltigi la tradukkostojn, ekzemple, per uzo de la nuntempaj disponeblaj teklologioj kiel aŭtomataj tradukiloj kaj nelokdependaj servoj.

Ĉapitro 1 donas ĝeneralan superrigardon pri la nuntempaj ŝanĝoj en la tradicia tradukmodelo, pri la dezirataj tradukaj kompetentecoj, pri la postulata engaĝiĝo de homoj en la maŝina tradukado kaj pri eblaj metodoj por mezuri la kvaliton de la tradukado.

Ĉapitro 2 diskutas la postulatajn ecojn de formatoj de dokumentoj taŭgaj por aŭtomata tradukado, koncerne la reuzon de sekcioj, kiuj samtempe ebligas precize kontroli historiajn ŝanĝojn.

Ĉapitro 3 studas la teknologiojn uzatajn en la sekvaj tradukmodeloj: Regulo-Bazita Maŝina Tradukado (RBMT), kiel la Distribuita Lingvo-Tradukado (DLT), Statistika Maŝina Tradukado (SMT), kaj Neŭrona Maŝina Tradukado (NMT).

Ĉapitro 4 analizas la prilaboron de la teksto en tri etapoj: kodiĝo al la unikoda kodaro, normaligo por malgrandigi la kvanton de la uzataj kodoj kaj la ĵetonigado por ke la teksto povu esti uzata en specifa tradukmodelo.

Ĉapitro 5 priskribas kelkajn tipajn uzojn de iloj uzataj en la tradukprocezo.

Ĉapitro 6 esploras kelkajn novajn metodojn kaj platformoj por malaltigi la koston de la tradukado.

¹¹ <https://www.911.gov/issues/ng911/>

¹² <https://eena.org/our-work/eena-special-focus/next-generation-112/>

1 Ŝanĝoj en la tradicia tradukmodelo

Por uzi novajn ilojn kaj teknikojn, oni devas kompreni kiujn novajn kapablojn ili proponas, kiel aliri ilin, kaj iliajn limigojn. La konstanta pliiĝo de la havebleco de aŭtomataj tradukiloj kaj malproksima aliro al grandegaj datenbankoj postulas novajn interagajn kapablojn.

En tiu ĉi ĉapitro oni diskutas ŝanĝojn en la kadro de kompetentecoj por tradukado de la projekto Eŭropa Magistra-nivela diplomo pri Tradukado, poste eblaj manierojn, kiel homoj povas engaĝiĝi en la maŝintradukado kaj fine kiel ŝanĝiĝas la formoj por mezuri la kvaliton de la homa kaj maŝina tradukado.

1.1 Kadro de kompetenteco por tradukado

La Ĝenerala Direkcio por Tradukado¹³ ([DGT](https://commission.europa.eu/about-european-commission/departments-and-executive-agencies/translation_en)) de la Eŭropa Komisiono¹⁴ havigas tradukon de verkitaj tekstoj al kaj el la dudek kvar oficialaj lingvoj de la Eŭropa Unio. DGT en partnereco kun pluraj universitatoj de eŭropaj kaj ekstereŭropaj landoj estigis en 2009 la projekton Eŭropa Magistra-nivela diplomo pri Tradukado ([EMT](https://commission.europa.eu/about-european-commission/departments-and-executive-agencies/translation_en)) per la eldono de sia kadro por tradukisto kaj tradukkompetenteco (Gambier 2009), kiu aljuĝas etiketon al altedukprogramoj, kiuj kontentigas la EMT-kvalitajn normojn por tradukisto-trejnado. Ĝi enhavas kvin kampojn de kompetenteco, el kiuj ĉi tiu studo analizas la teknologian aspekton.

En 2009 la teknikaj kompetentecoj estis:

- **Komputila lerteco:** Baza kompreno pri komputilaj operacioj, inkluzive de dosieradministrado kaj ĝenerala uzo de programoj.
- **Tradukiloj:** Scipovo pri traduk-rilata programaro, kiel komputila aŭtomatigita tradukiloj, sistemoj de tradukmemoroj kaj iloj por administri terminarojn.
- **Plurmediaj iloj:** Sperteco pri iloj por administri diversajn plurmediajn formatojn, inkluzive programojn por subtitolado kaj aŭdovida tradukado.
- **Informadika kaj komunikada teknologioj:** Lerteco en uzado de interret-bazitaj rimedoj kaj ciferecaj komunikiloj por esplorado kaj kunlaborado.
- **Solvo de Problemoj:** Kapablo identigi kaj solvi teknikajn problemojn, kiuj aperas dum la tradukprocezo.

La referenckadro por tradukisto kaj tradukkompetenteco de 2009 ricevis grandan modifon en 2017. Dum la bazaj principoj starigitaj en 2017 ankoraŭ staras, la kadro postulis ĝisdatigon en 2022 por reflekti la nunajn prioritatojn de eŭropaj tradukprogramoj; la kompetentecoj teknikaj ŝanĝis jene:

¹³ https://commission.europa.eu/about-european-commission/departments-and-executive-agencies/translation_en

¹⁴ https://commission.europa.eu/index_en

Teknologio (iloj kaj aplikaĵoj) inkluzivas ĉiujn sciojn kaj kapablojn uzatajn por efektiviĝi kaj konsili pri la uzo de nunaj kaj estontaj tradukteknologioj ene de la tradukprocezo. Ĝi ankaŭ inkluzivas bazan scion pri maŝintradukado-teknologioj kaj la kapablon efektiviĝi maŝintradukadon laŭ eblaj bezonoj.

Lernantoj (laŭ la EMT-kadro) devas scii kiel...

- Uzi la plej gravajn InformTehnologiajn (IT) aplikaĵojn, inkluzive de la plena gamo da oficejaj programoj, kaj rapide adaptiĝi al novaj iloj kaj IT resursoj, kritike taksii iliajn gravecojn kaj la efikon de ŝanĝo sur siaj laborpraktikoj;
- Fari efikan uzon de serĉiloj, korpuso-bazitaj iloj, tekstaj analiziloj, komputila tradukado kaj kvalifcertigoj kie estas konvene; Antaŭtrakti, prilabori kaj administri dosierojn kaj aliajn amaskomunikilarojn/fontojn kiel parto de la traduka laborfluo, ekz., interretajn paĝojn kaj plurmediajn dosierojn;
- Kompreni la bazojn de sistemoj por Maŝina Tradukado (MT) kaj ilian efikon al la tradukprocezo, kaj integri MT en traduklaborfluron kie estas konvene;
- Rekoni la gravecon kaj valoron de tradukado kaj lingvaj datenoj, pruvante datenlertecon;
- Apliki aliajn ilojn por subteno de lingvo- kaj tradukteknologio, kiel ekzemple laborflujajn administradilojn.

En 2009 la fokuso estis baza kompreno de komputilaj operacioj kaj, en aparta kategorio, solvo de problemoj, dum la kadro de kompetentecoj de 2022 pli amplekse kovras altnivelan scipovon kaj adapteblecon al teknologiajn ŝanĝojn. Ĝi ankaŭ inkluzivas emfazon en la efika uzo de interretaj rimedoj, komprenon kaj integriĝon de MT en la laborfluo, uzon de kvalifcertigoj kaj administradiloj.

En la sekvantaj ĉapitroj tiuj scioj estas analizataj provante trovi la bazajn teknikojn uzatajn.

1.2 Homa engaĝiĝo en maŝintradukado

En artikolo (Cornelius 2016) pri la kunlaboro de [FIT \(Federacio Internacia de Tradukistoj\)](#)¹⁵ kun la [QT21 \(angle: Quality Translation 21\)](#)¹⁶ projekto oni trovas diskuton, el la vidpunkto de FIT, pri kiel MT ŝanĝas la laboron de homoj en la metio de tradukado.

FIT estas internacia federacio de asocioj de tradukistoj, interpretistoj kaj terminologiistoj. Per aliĝo, pli ol 80 mil tradukistoj en 55 landoj sur la tera globo estas reprezentitaj en FIT. Resume, la celo de FIT estas antaŭenigi profesiecon en la disciplinoj, kiujn ĝi reprezentas. FIT estis partnero en la trijara QT21 projekto pri MT kiu funkciis de februaro 2015 ĝis februaro 2018.

¹⁵ <http://www.fit-ift.org/>

¹⁶ <https://cordis.europa.eu/project/id/645452>

Laŭ la opinio de FIT, en tiu artikolo, ekzistas sufiĉe da laboro por tradukistoj, kiuj ne sentas sin minacataj de MT. Sep el la diversaj manieroj kiel homoj, inkluzive de tradukistoj kaj ne-tradukistoj, nuntempe rilatas kun MT estas listigitaj kaj diskutitaj, jene:

1. **Provizo de enigo:** Tradukistoj devas provizi tekstojn, por la trejnado de la MT-sistemo, en la fontolingvo kaj responda tradukado en la cello. Tiu laboro inkluzivas:
 - Disdividi la tekston en segmentojn/frazojn, metante unu frazon po linio;
 - Paraleligi la frazojn por ke estu respondeco inter la originaj kaj tradukitaj segmentoj;
 - Forigi la metadatenojn, kiel grasajn tekstojn, ligilojn, ktp;
 - Normaligi la intervortajn spacetojn, citilojn, elekti ununuran formon de specialaj signoj kiam pluraj eblecoj ekzistas, ktp;
 - Ĵetonigi la tekston por apartigi unuojn uzatajn en la tradukmodelo. Ekzemple, kelkaj maŝintradukaj modeloj postulas, ke interpunkcioj estu apartigitaj de vortoj en la frazo, jene:

I'll take 3.5 of those, please.

al

I 'll take 3.5 of those , please .

- Ŝanĝi al minusklo aŭ majusklo kiel postulas la tradukmodelo. Kelkaj tradukmodeloj uzas majusklon nur por propraj nomoj, ne en la komenco de frazoj.
2. **Antaŭ prilaborado:** Por certigi legeblecon kaj tradukeblecon, la tradukendaj tekstoj kaj tekstoj konservitaj en la tradukmemoro prefere devas esti en kontrolita lingvo (tio estas, per aplikado de limigoj al leksikono, gramatiko kaj stilo) ĉar ĝi havas rimarkindan pozitivan efikon al la kvalito de la, maŝina tradukado.
 3. **Revizio de la maŝina tradukado:** Homoj asignas ĝeneralan signifon al kruda MT-produktaĵo kaj decidas ĉu plia pretigo estas necesa;
 4. **Postredaktado de la maŝina tradukado:** Korektado de eraroj en la kruda maŝintradukita teksto;
 5. **Uzo de proponitaj segmentoj de la MT-sistemo:** La maŝintradukado povas proponi opciojn el la tradukmemoro aŭ tradukita teksto kaj la tradukisto estas libera por uzi aŭ refuzi ilin;
 6. **Ne uzi la maŝinan tradukadon;**
 7. **Analiza taksado de la kvalito de la maŝina tradukado:** Tradukistoj povas taksi la MT enfokusigante la kadron de [MQM \(angle: Multidimensional Quality Metrics\)](https://themqm.org/)¹⁷, uzante normigitajn erarkategoriojn, prefere ol nur generante ununuran nombron indikantan ĝeneralan erarkategorion, kiel komplementon al, ne anstataŭaĵon por, metodoj, kiel BLEU (Papineni 2002), kiu estas vaste uzata por taksi la kvaliton de sistemoj uzataj por MT, sed ankaŭ kritikata (Lommel, 2016).

Oni konkludas, ke la MT-sistemoj ŝanĝas la manieron de laboro de tradukistoj, unuflanke ĝi disponigas rapidan tradukadon, kiu povas esti uzata kiel deirpunkto al pli bona fina verko adaptita al la dezirata publiko. Aliflanke, tradukistoj estas bezonataj por helpi en la kreado

¹⁷ <https://themqm.org/>

kaj taksado de MT-sistemoj adaptitaj al specifaj uzoj. Por fari ĉi tiun taskon, konstanta lernado pri la metodoj kaj limigoj de la MT-teknologioj estas bezonataj.

1.3 Kvalito de la tradukado

Du tute malsamaj aliroj estas uzataj por taksati la kvaliton de tradukado. Unu estas la analiza metodo, kiu inkluzivas la homan taksadon de tradukita teksto surbaze de antaŭdifinitaj kriterioj kaj tial reliefigas specifajn tradukproblemojn. Proponita kadro por taksati la kvaliton de tradukado estas la MQM (angle: *Multidimensional Quality Metrics*) evoluigita en la QT21 projekto de la Eŭropa Komisiono en 2015. Ĉi tiu kadro estas uzata por identigi erartipojn ene de sep dimensioj: terminologio, precizeco, lingvistikaj konvencioj, stilo, lokaj konvencioj, taŭgeco por la publiko, kaj dezajno. Tiu unua versio de MQM estas la bazo por daŭranta ASTM projekto nomita “*New Practice for Analytic Evaluation of Translation Quality*”¹⁸.

La dua estas la sinteza metodo, kiu donas gradon por la tutaĵo de la tradukado kaj estas generita per aŭtomata proceduro sen homa interveno. Unu el la aliroj estas normigota en daŭranta ASTM projekto nomita “*New Practice for Holistic Quality Evaluation System for Translation*”¹⁹. Ĉi tiu metodo inkluzivas perceptadon de la materialo kiel tuto, sentemon al specifaj detaloj, fidelecon de signifo-transsendo, kuntekston, literaturan impreson, legeblecon, tekstotipon, celon, intencitan publikon, lingvajn trajtojn, kaj aliajn faktorojn.

Alia tipo de la sinteza metodo estas taksado de la kvalito de MT-sistemoj. La tradicia aliro per la homaj juĝadoj por determini precizecon, fluecon, kaj adekvatecon estas multekosta kaj postulas tempon, pro tio ĝi ne kongruas kun la postulo taksati la kvaliton de MT-modelo en mallonga tempo kaj malalta kosto. Por solvi ĉi tiun problemon, pluraj maŝinaj tradukaj mezuroj, kiuj povas esti aŭtomate generitaj de komputilo, estis proponitaj. Por taksati tradukon de fontolingvo al cellingvo, tiuj mezuroj uzas hom-generitan korpuson de tekstosegmentoj, kiu enhavas la fontan tekston kaj unu aŭ pli da referencaj tradukoj en la cellingvo. Ĉiu mezuro uzas aron da reguloj por kompari la maŝin-generitan tradukon kun la referencaj tradukoj kaj provizas gradon, kiu esperinde havas altan korelacian kun la homa traduka juĝo. Kelkaj el la unue uzataj mezuroj kiel WER, PER kaj TER baziĝas sur la Levenshtein-distanco (Levenshtein 1966). Aliaj mezuroj kiel BLEU, NIST kaj METEOR komparas la aperon de vortoj kaj trovas pli grandan aplikeblecon en la takso de Maŝina Tradukado.

- **WER** (angle: *Word Error Rate*) mezuras la nombron da enmetoj, forigoj, kaj anstataŭigoj necesaj por transformi la maŝin-generitan rezulton al unu el la referencaj tradukoj. Ĝi estas vaste uzata en voĉrekono sed ankaŭ povas esti aplikata al maŝina tradukado.

WER estas simpla kaj ĉiam provizas la saman mezuron pri kiom da ŝanĝoj necesas por transformi la maŝinan tradukadon en la referencan tradukadon. Al tiu mezuro

¹⁸ www.astm.org/workitem-wk46396

¹⁹ <https://www.astm.org/workitem-wk54884>

mankas semantika taksado, ĉar WER ne konsideras semantikan precizecon aŭ fluecon, efektive ĝi pure mezuras la distancon de la redaktadoj.

- **PER** (angle: *Position-independent Error Rate*) estas projektita por esti malpli sentema al la pozicio de eraroj. PER provizas mezuron de eraro kiu ne dependas de la pozicio de vortoj, igante ĝin utila por taksado tradukojn kie la vortordo povas varii. Simile al WER, PER ne konsideras semantikan signifon aŭ tradukan fluecon.
 - **TER** (angle: *Translation Edit Rate*) mezuras la nombron da redaktadoj (enmetoj, forigoj, anstataŭigoj, kaj ŝanĝoj) necesaj por transformi maŝin-generitan tradukon al referenca traduko. Ĉar TER ankaŭ konsideras ŝanĝojn en vortordo, ĝi provizas pli nuancitan taksadon ol WER kaj ofertas enrigardojn pri la flueco de la traduko. La kalkulado de TER postulas pli kompleksan algoritmon kompare kun BLEU kaj WER.
- BLEU** (angle: *Bilingual Evaluation Understudy*) (Papineni 2002) fariĝis normo en MT-taksado kaj estas vaste rekonata en la esplora komunumo. Unu el la plej ofte uzataj mezurkaj, BLEU, estis proponita de IBM kaj estas vaste uzata ĉar ĝi povas provizi tujajn rezultojn kaj gvidon en MT-esplorado kaj montris altan korelacian kun homa traduka juĝo. Ĉi tiu propono komparas la rezulton de MT kun referenca traduko laŭ la statistikoj de mallongaj sekvencoj da n vortoj (nomitaj vortaj n -gramoj, tipe enhavantaj inter 1 kaj 4 vortoj). Ju pli da tiuj 1-gramoj, ke la traduko havas komune kun la referencaj tradukoj, des pli bona estas juĝita la traduko. La studo (Papineni 2002) montris fortan korelacian inter tiuj aŭtomate generitaj gradoj kaj homaj juĝoj de traduka kvalito.

Taksado kiu uzas n -graman kunokazo-statistikon postulas korpuson por taksado en la fontolingvo kune kun unu (aŭ prefere pli) altkvalitaj referencaj tradukoj. La gradado tiam povas esti farita per komparado de la frakcio de n -gramoj en la MT, kiuj ankaŭ okazas en la referencaj tradukoj. N -grama kunokaza gradado estas tipe efektivigita segmento post segmento, kie segmento estas la minimuma unuo de traduka kohero, kutime unu aŭ kelkaj frazoj. La n -grama kunokaza statistiko, bazita sur la aro da n -gramoj tradukitaj per la MT kaj referencaj segmentoj, estas kalkulata por ĉiu el tiuj segmentoj kaj post akumulata inkludante ĉiujn segmentojn. Antaŭ gradado, la tradukita teksto estas normaligita kaj ĵetonigita por plibonigi la efikecon de la gradada algoritmo. Ĉi tiu antaŭ-preparado estas aplikata al ambaŭ la traduko, por esti gradita, kaj al la referencaj tradukoj. Ekzemple, en la angla la sekvanta preparado estas aplikata:

- Usklecaj informoj estas forigitaj. Tekstoj estas konvertitaj al minusklaj aŭ majuskulaj signoj kiel postulas la tradukmodelo;
- Numeraj informoj (en formo de sekvencoj da ciferoj, komoj kaj punktoj) estas konservitaj kiel unuopaj vortoj;
- Interpunkcio estas ĵetonigita en apartajn vortojn (escepte de streketoj kaj apostrofoj);
- Najbaraj ne Askiaj vortoj (kiuj okazas kiam ne angla fontoteksto estas transferita al la tradukita teksto) estas kunigitaj en unuopajn vortojn.

Unu el la kaŭzoj de ĝia sukceso estas ĝia simpleco por kalkuli kaj facileco por efektiviĝi. Pro tio ke BLEU ĉefe mezuras n-graman precizecon, ĝi ne rekte konsideras semantikan signifon aŭ fluecon. Ankaŭ, la kvalito de BLEU-grado dependas forte de la nombro kaj kvalito de la referencaj tradukoj.

- **NIST** (angle: *National Institute of Standards and Technology*) taksis la n-graman gradadon de BLEU (Doddington, 2002) laŭ korelacio kun homaj taksadoj, sentiveco, kaj konsistenco. Ĉi tiu artikolo ankaŭ proponas ŝanĝojn por plibonigi fluecon kaj adekvatecon kaj taksis ĝian efikecon laŭ fontolingvo, nombro da referencaj tradukoj, grandeco de la segmentoj, pli da lingva trejnado kaj konservado de ukleco. NIST estas plivastigo de BLEU, kiu enkondukas pezojn por n-gramoj surbaze de ilia informiĝeco. Ĉi tiu aliro celas pli bone kapti la kvaliton de tradukoj per emfazo de la signifo de malpli oftaj sed pli informaj n-gramoj. La kalkulado de NIST estas pli komplika ol tiu de BLEU, pro la pezoj de n-gramoj. La mezuro ankaŭ dependas de la kvalito kaj diverseco de la referencaj tradukoj.
- **METEOR** (angle: *Metric for Evaluation of Translation with Explicit Ordering*) (Banerjee 2005)
Ĉiu ebla kongruo inter la referenco kaj la MT estas gradita surbaze de kombino de pluraj karakterizaĵoj. Ĉi tiuj nuntempe inkluzivas unigraman precizecon, unigraman rememoron, kaj rekta mezuro de kiom ne ordigitaj estas la vortoj de la MT rilate al la referenco. METEOR celas trakti kelkajn mankojn de BLEU per enkalkulado de sinonimoj, devenradiko de vorto, kaj vortordo. Ĝi kombinas multajn taksatajn komponantojn inkluzive de precizeco, rememoro, kaj sinonima kongruo. METEOR enkorpigas sinonimojn kaj devenradikojn, permesante al ĝi taksi tradukojn surbaze de semantika simileco anstataŭ ekzaktaj kongruoj. Ĝi konsideras vortordon, farante ĝin pli robusta en taksado de fluecaj tradukoj. La kalkulado de METEOR estas pli kompleksa ol tiu de BLEU, implicante multajn etapojn de analizo. Kiel BLEU, la efikeco de METEOR povas esti influita de la kvalito kaj nombro da referencaj tradukoj.

La plej ofte uzataj aŭtomataj mezurmetodoj por taksi MT estas BLEU kaj METEOR. Dum ĉi tiuj metodoj estas rapidaj, malmultekostaj, kaj reprodukteblaj, ili ne donas pliajn informojn pri la naturo de la tradukproblemoj. Aldone, Lommel (2016) faris eksperimentojn por taksi la efikecon de la BLEU-mezuro, kiam ĝi taksas traduk-kvaliton. Li pridubas la precizecon de BLEU rimarkante ĝiajn limigojn en fidele reprezenti la homan komprenon pri traduk-kvalito. Li argumentas, ke BLEU, kiu mezuras similecon de segmentoj de tekstoj al referenca traduko, eble ne fidinde indikas sisteman efikecon de la MT pro ĝia natura malprecizeco kaj dependeco de ununura referenco. Ĉi tio povas fari ĝin malbonan mezurilon de reala traduk-kvalito. Dum BLEU estas kohera kaj utila en kunteksto de disvolviĝo de MT, ĝi malsukcesas provizi detalajn komprenojn pri specifaj tradukaj problemoj.

Ambaŭ tipoj de mezuroj, kiel analizaj kaj sintezaj mezuroj, havas siajn rolojn en traduka taksado, kun sintezaj aŭtomataj mezuroj estante rapidaj sed malpli informplenaj, kaj analizaj mezuroj ofertante detalan analizon de la problemoj kaj gvidadon por plibonigo.

2 Taŭga formato por la dokumentoj

La plej taŭga formato por konservi informojn dependas de la fina uzo de la dokumento. En la nuna studo la postulata deirpunkto estas trovi formaton kiu ebligas dokumentojn taŭgajn por aŭtomata tradukado, kaj por reuzo de sekcioj, kiuj samtempe ebligas precize kontroli historiajn ŝanĝojn, kaj estas facile ŝanĝeblaj per komputilaj programoj.

Enigo de tradukenda informo estas ĉiam konvertita al la formo de teksto antaŭ ol ĝi estas sendita al aŭtomata tradukilo, ankaŭ estas pli facile kompari tekstojn ol binarajn datenojn. Pro tio la preferataj formatoj estas tiuj kiuj konservas informojn kiel tekston.

La dokumentoj devas ankaŭ esti strukturitaj por ebligi enhavi metadatenojn, kiuj priskribas la version, la lingvon, la formaton por prezentado, la lokon kie troviĝas aldonaj dosieroj, ktp.

En ĉi tiu ĉapitro, [XML](#)²⁰-bazitaj formatoj de dokumentoj estas ekzamenitaj, ĉar ili estas kutime uzataj por kontroli la tradukadon de dokumentoj. Aldone la programo [GIT](#)²¹ estas prezentata kiel ebla ilo por kontroli la historiajn ŝanĝojn de la dokumentoj uzataj en la tradukado.

2.1 Formatoj de dosieroj por komputilaj tradukprogramoj

Konstanta ŝanĝiĝo kaj plurforma prezentado estas nuntempa trajto de disvastiĝo de informoj. Multnaciaj entreprenoj uzas specifajn formatojn kaj ilojn por adaptigi siajn dokumentojn kaj programojn al la specifaj lingvaj kutimoj de divesaj landoj. Ekzemple, teknika dokumento, kiu priskribas la uzon de komunikilo kutime estas prezentata almenaŭ en du malsamaj formatoj: kaj kiel TTT-paĝo kaj kiel printebla PDF. Ankaŭ, ene de tiu dokumento, informoj kiel nombroj, datoj, nomoj plurfoje povas esti programaj variabloj determinitaj dum la prezentado en la ekrano, pro tio, ili ne estas konataj de la tradukisto. Aliflanke, la tradukprocezo devas plurfoje ampleksi limigojn pri tekstolongo, ĝustan lokiĝon de metadatenoj (ekzemple: informo ke parto de la teksto estas grasa, kursiva aŭ substrekita), lokiĝon en la teksto de bildoj, kaj piktogramoj, kiuj ne ĉiam estas apartaj de la fina tradukita teksto, sed estas konservitaj en apartaj dosieroj, ktp.

Por difini tiujn elementojn ekzistas kelkaj normigitaj formatoj de la fontodokumentoj el kiuj la celdokumentoj en iu ajn el pluraj specifaj formatoj (TTT-paĝo, PDF, ktp) povas esti kreitaj. Du eblaj kutime uzataj formatoj, bazitaj en la XML formato, estas la [DocBook](#)²² kaj [DITA](#)²³ (angle: *Darwin Information Typing Architecture*) ambaŭ normigitaj de OASIS (angle: *Organization for the Advancement of Structured Information Standards*). Vidu en **Figuro 1** tekston kun tri metadatenoj: 1. Enmetita piktogramo, el la dosiero “`figures/ons_ond.gif`”; 2. Teksto en la


²⁰ <https://www.w3.org/TR/REC-xml/>

²¹ <https://git-scm.com/>

²² <https://docbook.org/specs/docbook-5.2-spec-os.html>

²³ <http://docs.oasis-open.org/dita/dita/v1.3/dita-v1.3-part3-all-inclusive.html>

grasa formato, indikata per “<uicontrol>...</uicontrol>”; 3. Teksto en la grasa kaj substrekita formatoj, indikata per “<u><uicontrol>...</uicontrol></u>”.

<p>Step by Step</p> <p>1) Click  in the header bar. The settings menu opens.</p> <p>2) Select a device under Incoming calls and under <u>Outgoing calls</u></p>
<pre><steps><step><cmd> Click <image href="figures/ons_ond.gif" height="16.680pt" width="16.680pt" placement="inline"/> in the header bar. </cmd><stepresult><p> The settings menu opens. </p></stepresult></step><step><cmd> Select a device under <uicontrol>Incoming calls</uicontrol> and under <u><uicontrol>Outgoing calls</uicontrol></u></cmd></step></steps></pre>

Figuro 1 Celdokumento en la PDF formato kaj fontodokumento en la DITA formato.

Pro tio ke la DITA formato estas bazita en la XML-formato, kelkaj signoj de la fontoteksto devas aperi koditaj en la XML-dokumento, ekzemple, la signo “<”, laŭ la XML normo, devas aperi kodita kiel “<”. Vidu ekzemplon en la **Figuro 2**. La sama signo povas aperi ankaŭ kodita per Unikoda kodopunkto kiel “<” aŭ “<”; aldonaj klarigoj pri tiuj kodoj troviĝas en la sekcio 4.1.

<p>Specifications with varying content <user name></p>
<pre><entry colname = "1" colsep = "0"> Specifications with varying content </entry><entry colname = "2"> &lt;user name> </entry></pre>

Figuro 2 Koditaj signoj kiel “<” povas aperi en la XML dosiero.

Por la tradukado de la fontodokumento al alia lingvo la dosieroj en la DocBook aŭ DITA dosierujo estas unue konvertitaj al dosieroj en la formato [XLIFF-1.2²⁴](https://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html) aŭ [XLIFF-2.1²⁵](http://docs.oasis-open.org/xliff/xliff-core/v2.1/os/xliff-core-v2.1-os.html) (angle: *XML Localization Interchange File Format*). Oni trovas diskuton pri la bezono kaj avantaĝoj de tiuj normigitaj formatoj en “*Why XLIFF and Why XLIFF 2?*” (Filip 2016). Post la tradukado nova dosierujo en la origina formato (DocBook aŭ DITA), nun en la cellingvo, estas kreita per la tradukilo.

Tradukiloj traktas, en la formato XLIFF, la tekston, kiu devas esti tradukita. En tiu formato, ankaŭ bazita sur XML, la teksto estas dividita en segmentojn, kiuj kutime enhavas unu aŭ kelkajn frazojn. La segmentoj estas individue tradukitaj kaj post kontrolitaj kaj aprobitaj.

Figuro 3 montras XLIFF-dokumenton kun tri segmentoj, la unua estas tradukita kaj aprobita,

²⁴ <https://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html>

²⁵ <http://docs.oasis-open.org/xliff/xliff-core/v2.1/os/xliff-core-v2.1-os.html>

la dua estas tradukita sed ne aprobita kaj la tria ne estas ankoraŭ tradukita. La metadatenoj "`<target state=...`" indikas la staton de la segmento.

```
<?xml version="1.0" encoding="UTF-8"?>
<xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2">
  <file original="example.txt" source-language="en" target-language="pt"
    datatype="plaintext">
    <body>
      <trans-unit id="1" approved="yes">
        <source> Usage Restrictions</source>
        <target state="final">Restrições de Uso</target>
      </trans-unit>
      <trans-unit id="2" approved="no">
        <source> History of Changes</source>
        <target state="translated">Histórico de Alterações</target>
      </trans-unit>
      <trans-unit id="3" approved="no">
        <source>Getting Started</source>
        <target state="needs-translation"></target>
      </trans-unit>
    </body>
  </file>
</xliff>
```

Figuro 3 Simpla dokumento en la formato XLIFF enhavanta 3 segmentojn.

Dum la tradukado, oni konsultas memoron de antaŭaj tradukitaj segmentoj kaj terminarojn. La tradukisto povas ankaŭ konservi la novajn tradukojn en Memoro de Tradukado (MT) kaj konservi novajn terminojn en terminaro.

MT estas kutime konservita en la [TMX²⁶](#) (angle: Translation Memory eXchange) formato, ankaŭ bazita sur XML. La MT enhavas la segmentojn en la fontolingvo kaj kutime iliajn tradukojn en uno cello, ili ankaŭ povas esti tradukitaj al pluraj lingvaj variantoj, vidu ekzemplon de MT en **Figuro 4**.

```
<?xml version="1.0" encoding="utf-8"?>
<tmx version="1.4">
  <header creationtool="Smartcat" creationtoolversion="7.0.0.0"
    segtype="sentence" o-tmf="ATM" adminlang="en-US" srclang="en"
    datatype="plaintext">
  </header>
  <body>
    <tu>
      <tuv xml:lang="en"> <seg>Usage Restrictions</seg> </tuv>
      <tuv xml:lang="pt-BR"> <seg>Restrições de Uso</seg> </tuv>
    </tu>
    <tu>
      <tuv xml:lang="en"> <seg>History of Changes</seg> </tuv>
      <tuv xml:lang="pt-BR"> <seg>Historico de Alterações</seg> </tuv>
    </tu>
  </body>
</tmx>
```

Figuro 4 Dokumento en la formato TMX enhavanta 2 tradukunuojn.

²⁶ <https://www.gala-global.org/tmx-14b>

Terminaro estas kutime konservita en uno el la dialektoj de la formato [TBX](#)²⁷ (angle: *TermBase eXchange*), kiuj ankaŭ baziĝas sur XML. La baza dialekto nomata TBX-Core estas priskribita de ISO Standard 30042:2019. Estas nur du devigaj datenkategorioj en TBX: termino kaj lingvo. Pluraj el la ceteraj datenkategorioj, inkluzive de difino, kunteksto, vortparto, kaj temokampo estas tre gravaj kaj devus esti inkluditaj en la terminaroj kiam ajn eblas. Vidu ekzemplon de Terminaro en **Figuro 5**.

En resumo, uno el la plej gravaj formatoj de dosieroj uzataj dum la tradukado estas la formato XLIFF, kiu estas prezentata al la tradukisto, per la tradukilo, kiel segmentoj de tekstoj. Antaŭ la tradukado la fontodokumento devas esti konvertita al la formato XLIFF, kaj post la tradukado la dokumento devas esti konvertita al la origina formato.

Dum la tradukado MT kaj Terminaro estas uzataj por provizi proponojn de eblaj tradukoj. Hodiaŭ, en aŭtomataj tradukiloj, la unua propono de tradukado estas bazita en la MT, terminaro kaj ankaŭ en modeloj de Artefarita Inteljekto (AI). Homa tradukado estas la dua etapo kaj postredaktado la tria etapo. Tiuj etapoj estas konservitaj en la segmentoj de la dosiero XLIFF per la metadatenaj statoj:

- `<target state="needs-translation">...</target>`
- `<target state="translated">...</target>`
- `<target state="final">...</target>`
- `<trans-unit id="1" approved="yes">`
- `<trans-unit id="2" approved="no">`

Pluraj formatoj de dokumentoj ekzistas; la formato DITA estas taŭga elekto por dokumentoj organizitaj en recikleblajn temojn prefere ol liniaj dokumentoj. Ĉi tiu modula aliro permesas enhavrezon el ununura fonto. DocBook sekvas tradician, dokumentstrukturon, kiu inkludas ĉapitrojn, sekciojn, apendicojn, ktp, pro tio ĝi estas taŭga por hierarkiaj dokumentoj.

Krom la formato XLIFF uzata por traduki tekstojn, aliaj formatoj estas uzataj por specifaj cirkonstancoj. Ekzemple, programoj de la [GNU Linuksa](#)²⁸ operaciumo estas disponigitaj tradukitaj al pluraj lingvoj per la uzo de tradukitaj dosieroj en la formato Portebla Objekto (PO) kun la dosiersufikso ".po". Ĉi tiu formato ebligas ke, sen kodoŝanĝo, programoj povas prezenti programe ŝanĝeblajn mesaĝojn tradukitaj al pluraj lingvoj. Tradukiloj kiel [Lokalize](#)²⁹, [PoEdit](#)³⁰, [Virtaal](#)³¹, ktp estas uzataj en la [Traduko-Projekto \(TP\)](#)³² de GNU por krei la dosierojn uzatajn en la linuksaj programpakaĵoj.

²⁷ https://www.terminorgs.net/downloads/TBX_Basic_Version_3.1.pdf

²⁸ <https://www.gnu.org/>

²⁹ userbase.kde.org/Lokalize

³⁰ <https://poedit.net/>

³¹ <http://www.virtaal.org/>

³² <https://translationproject.org/>

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tbx SYSTEM "http://www.oasis-
open.org/committees/tc_home.php?wg_abbrev=tbx">
<tbx xmlns="http://www.oasis-
open.org/committees/tc_home.php?wg_abbrev=tbx">
  <header>
    <fileDesc>
      <title>Example TBX File</title>
      <source>Example Source</source>
      <date>2024-08-10</date>
    </fileDesc>
  </header>
  <body>
    <termEntry>
      <ntigTerm>
        <langSet lang="en">
          <tig>
            <term>
              <form>computer</form>
              <termNote type="definition">An electronic device for storing
and processing data.</termNote>
            </term>
          </tig>
        </langSet>
        <langSet lang="fr">
          <tig>
            <term>
              <form>ordinateur</form>
              <termNote type="definition">Appareil électronique pour le
stockage et le traitement des données.</termNote>
            </term>
          </tig>
        </langSet>
      </ntigTerm>
    </termEntry>
  </body>
</tbx>

```

Figuro 5 Terminaro enhavanta terminojn kaj ĝiajn difinojn.

2.2 Kontrolo de historiaj ŝanĝoj

Elekti tekstan formaton por la dosieroj de dokumentoj estas grava unua paŝo, tamen, spuri ŝanĝojn en la dokumentaro tra la tempo estas nemalhavebla metodo por garantii la kvaliton de la originalaj kaj tradukitaj dosierversioj. Ĉiuj publikigitaj versioj de la dokumentoj estas gravaj, ĉar ili enhavas la intelektan proprieton de la entrepreno, krome, aliaj gravaj aldonaj informoj estas: kiuj ŝanĝoj estis faritaj, kiu faris ilin, kaj kiu aprobis ilin, ĉar tiuj ŝanĝoj povas havi konsekvencojn en la uzo de la produkto same kiel jurajn konsekvencojn.

Diskuto pri la graveco de la uzo de laborfluo de revizioj integrita en la tradukilo, dum trejnado de studentoj, troviĝas en Gledhill 2019. La propono en ĉi tiu ĉapitro estas por kontrolo ankaŭ ekstere el la tradukilo per GIT.

Tradukiloj eble havas siajn proprajn sistemojn por kontroli versiojn, tamen, se la entrepreno uzas eksterajn tradukservojn por fari siajn tradukojn, ĝi devus uzi sian propran sistemon por spuri la ŝanĝojn en la dokumentoj, inkluzive ŝanĝoj en la tradukmemoro kaj terminaro.

Hodiaŭ la plej ofte uzata sistemo por kontroli versiojn estas GIT, libera programaro kiu origine estis uzata de programistoj, sed nun estas uzata en multaj aliaj areoj. Multaj interretaj retejoj kiel GitHub³³, GitLab³⁴, aŭ Bitbucket³⁵ uzas GIT por konservi projektojn. En ĝiaj deponejoj kutime estas ankaŭ dokumentoj konservitaj en teksta formato same kiel programkodo. Tial, eblas konservi la fontodokumentojn, se ili estas en formato bazita sur XML, por ke ili estu kontrolita, same kiel oni konservas la koncernan kodon uzatan por generi la finajn dokumentojn antaŭ ili estas publikigitaj. GIT ne estas intencita por kontroli la versiojn de binaraj dosierformatoj, malgraŭ tio, ekzistas GIT-versio uzata por konservi grandajn dosierojn. Por solvi ĉi tiun problemon, estis kreita solvo nomata GIT LFS³⁶ (angle: *Large File Storage* – granda dosiera deponejo). Esence, anstataŭ konservi la dosieron mem en la deponejo, GIT LFS simple konservas indikilon al kie tiu dosiero vere troviĝas.

GIT havas kuriozan historion³⁷: ekde 2002 Linus Torvalds uzis senpagan licencon de komerca sistemo por kontroli la evoluon de Linukso; sed, en 2005, la posedanta kompanio nuligis la senpagan licencon uzatan de la programistoj de Linukso. Ĉi tio instigis la Linuksan komunumon (kaj precipe Linus Torvalds, la kreinton de Linukso) por evoluigi sian propran ilon surbaze de kelkaj el ilia antaŭa sperto. Rezulte, en kelkaj monatoj Torvalds evoluigis sian propran sistemon kiu estas rapida, simpla, distribuita, kapabla trakti grandajn projektojn efike kun ampleksa subteno al paralela disvolviĝo de programoj. La dokumentado de GIT donas plurajn signifojn por ĉi tiu akronimo, “*Global Information Tracker* – tutmonda informo-spurilo” se vi estas en bona humoro aŭ malagraba, stulta, malestimata kaj malbona laŭ brita angla slango. Hodiaŭ GIT estas uzata de pli ol 90% de la programistoj kaj estas trovita kiel maniero kontroli projektojn, kiuj ŝanĝiĝas tra la tempo en multaj malsamaj areoj.

GIT estas distribuita sistemo desegnita por spuri ŝanĝojn en fontokodo kaj aliaj tipoj de tekstaj dosieroj. En la specifa kampo de tradukado, la gravaj dosierformatoj, krom kodaj dosieroj, kiuj povas esti uzataj kun GIT estas:

- Agordaj dosieroj: Tekstobazitaj agordaj kaj datenaj dosieroj kiel datenaranĝo CSV “.csv” (angle: *Comma-Separated Values* – perkome disigitaj valoroj), JSON “.json” (angle: *JavaScript Object Notation*), YAML “.yaml”, “.yml” (angle: *Yet Another Markup Language*), kaj INI “.ini” (angle: *Initialization*);
- Dosieroj de dokumentoj: Simplaj dosierformatoj uzataj por dokumentoj kiel *Markdown* “.md” kaj simpla teksto “.txt”;
- HTML/CSS: Dosieroj uzataj en retpaĝoj, kiel HTML “.html” (angle: *Hypertext Markup Language*) kaj CSS “.css” (angle: *Cascading Style Sheets*);
- XML: Dosieroj “.xml” (angle: *Extensible Markup Language*) uzataj por agorda kaj datena interŝanĝo.

³³ <https://github.com/>

³⁴ <https://about.gitlab.com/>

³⁵ <https://bitbucket.org/>

³⁶ <https://git-lfs.com/>

³⁷ <https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git>

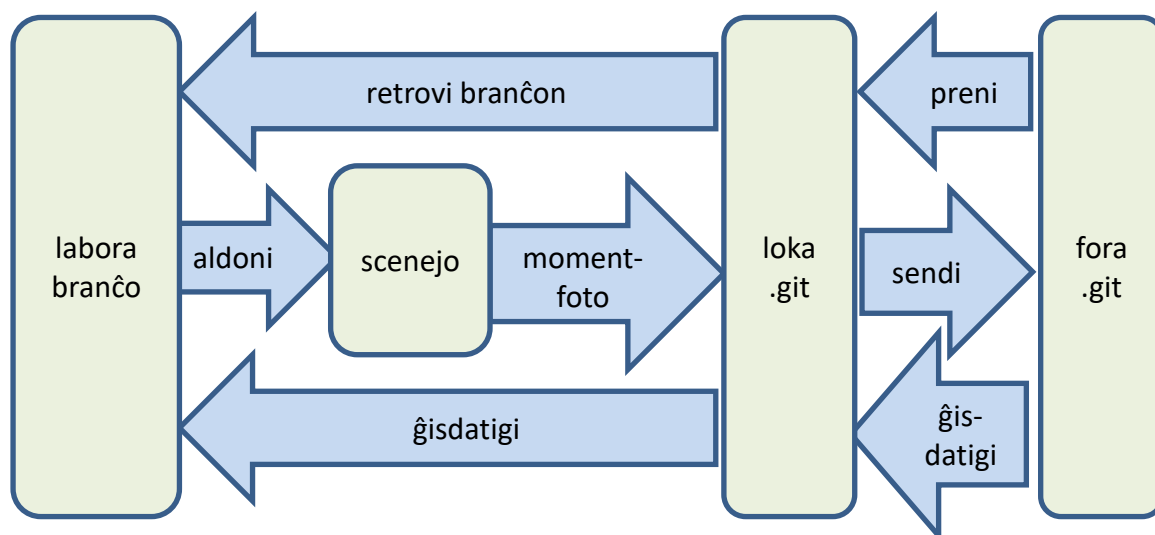
GIT ankaŭ povas spuri grandajn binarajn dosierojn se ili estas konservitaj en aparta dosierujo. Iuj el tiuj dosieroj estas:

- Bildoj: Dosieroj “.jpg, .png, .gif”;
- Binariaj dosieroj: Ekzekuteblaj dosieroj “exe, .bin” kaj aliaj tradukitaj binaraj dosieroj;
- Arkivoj: Kunpremitaj dosieroj “.zip, .tar, .gz”.

Ĉiu uzanto havas, en sia labora dosierujo, kompletan kopion de la deponejo, en dosierujo nomita “.git”, inkluzive ĝian historion. Ĉi tio ebligas lokan laboron, en specifa branĉo, sen reta aliĉo, kaj permesas uzantojn labori sendepende de centra servilo. Samtempe ĉiuj deponejoj de GIT funkcias kiel sekurkopio de la tuta projekto.

Ĉiuj dosieroj de projekto, kiu estas konservita sub GIT-kontrolo, estas lokitaj en dosierujo, konata kiel deponejo (angle: *repository*).

Kiam oni laboras en loka dosierujo, post ŝanĝo en dosieroj, la sekvajn agojn devas esti plenumitaj por sendi momentfoton de la ŝanĝoj al la GIT deponejo, vidu en Bildo 2-1:



Bildo 2-1 Eblaj agoj en GIT.

- **Aldoni** dosierojn al scenejo, nome prepari por momentfoto (angle: *add*);
- Preni **momentfoton** (angle: *commit*);
Ĉi tiu momentfoto havas identigilon kaj enhavas metadatenojn kiel la aŭtoron, tempomarkon, kaj ŝanĝosendo-mesaĝon.
- **Sendi** momentfoton al fora deponejo (angle: *push*);
- **Preni** momentfoton de la fora deponejo al la loka deponejo (angle: *fetch*);
- **Retrovi branĉon** de la loka deponejo al la labora dosierujo (angle: *checkout*);
- Aliaj uzantoj povas **ĝisdatigi** siajn lokajn deponejojn kaj labora dosierujo, tio estas, plenumi ambaŭ agoj: preni la momentfoton el la fora deponejo kaj retrovi branĉon (angle: *pull*).

Pluraj programistoj povas labori paralele, en deponejoj konservitaj en malsamaj maŝinoj, samtempe en la sama aŭ en malsamaj branĉoj. Ĉi tiuj branĉoj povas post esti kunfanditaj al la ĉefversio, ebligante paralelan laborfluon kaj eksperimentadon ne ŝanĝante la ĉefan dokumenton.

GIT povas montri la historion de iu ajn branĉo kaj reveni al pli fruaj versioj de dokumento se necese. Ankaŭ eblas montri la diferencojn inter du dosieroj aŭ inter du versioj de dosieroj.

Resume, GIT estas taŭga ilo por konservi dosierojn kaj spuri ŝanĝojn ene de projekto, kiuj enhavas nur tekstajn dosierojn, aŭ enhavas binarajn dosierojn konservitajn en aparta dosierujo. GIT provizas metodojn por kontroli konformecon kaj certigi kvaliton, certigante ke ŝanĝoj estas dokumentitaj kaj spureblaj. Ĉiu ŝanĝosendo en GIT reprezentas momentfoton de la dokumento ĉe specifa tempo. Ĉi tio signifas ke ajna historia versio de la dokumento estas alirebla, reviziebla aŭ restaŭrigebla laŭbezono.

3 Maŝina Tradukado

Pluraj provoj por aŭtomate traduki tekstojn estis faritaj ekde la komenco de la uzo de komputiloj. Por pli boni kompreni tiujn provojn oni povas dividi ilin, laŭ la teknologio kaj procedo uzata, en kelkajn fazojn, jene:

1950-aj ĝis 1980-aj jaroj Regulo-Bazita Maŝina Tradukado (RBMT):

La plej fruaj provoj pri maŝina tradukado baziĝis sur regulo-bazitaj sistemoj, kie lingvistoj kaj inĝenieroj kreis ampleksajn arojn da lingvaj reguloj kaj dulingvajn vortarojn.

La Georgetown-IBM eksperimento en 1954 (Hutchins 2004) estas deirpunkto, kie estis montritaj tradukoj de pli ol sesdek rusaj frazoj al la angla uzante trukartojn por enigo de teksto kaj presita eligo.

Unu grava modelo de ĉi tiu tipo estas la unua fazo de Distribuita Lingvo-Tradukado (DLT).

- **1990-aj ĝis 2010-aj jaroj Statistika Maŝina Tradukado (SMT):**

La apero de la ideo pri Statistika Maŝina Tradukado (SMT) estas atribuita al la publikigo de memorando (Weaver 1949), simple nomita “*Translation*” (Tradukado). En ĉi tiu memorando, Weaver proponis apliki statistikajn metodojn, precipe tiujn uzatajn en kriptografio dum la Dua Mondmilito, al la problemo de maŝina tradukado. Kun la apero de pli prilabora povaj komputiloj kaj aliro al grandaj korpusoj de dulingvaj tekstoj, SMT aperis kiel ĉefa metodo por tradukado.

- **2010-aj ĝis nuna tempo Neŭrona Maŝina Tradukado (NMT):**

La plej moderna aliro estas uzi profundan lernadon por modeligi la tutan tradukprocezon per unuopa artefarita neŭrona reto. Sistemoj de tradukado per NMT uzas neŭronajn retojn por aŭtomate lerni la kompleksajn rilatojn inter lingvoj el granda datenaro.

La modeloj antaŭdiras la vicon de la tradukitaj vortoj en la cellingvo surbaze de la enigo en la fontolingvo. Tipe ili produktas pli fluecajn kaj naturajn tradukojn, precipe por pli longaj frazoj kaj kompleksaj strukturoj. Tamen, ili postulas konsiderindajn komputilajn rimedojn kaj grandan datenaron, ankaŭ, ili povas barakti traktante nuancojn de signifo, kiu dependas de la kunteksto, aŭ lingvoj kiuj havas malmultajn disponeblajn rimedojn.

Kompreneble, aliaj metodoj kiuj kunigas plurajn malsamajn teknikojn estis kreitaj por kombini elementojn de regulo-bazita, statistika, kaj neŭrona aliroj por utiligi la avantaĝojn, ke ĉiu el ili provizas.

Hibridaj sistemoj povas uzi regulojn por trakti specifajn lingvajn fenomenojn dum ili fidus je statistikaj aŭ neŭronaj metodoj por aliaj partoj de la tradukprocezo.

Resume, pli altnivelaj kaj precizaj metodoj fariĝis eblaj nur post kiam la necesaj kapabloj estis haveblaj. La kresko de memorkapablo kaj prilabora povo de komputiloj, same kiel la kreado

de grandaj korpusoj kaj la havebleco de granda kvanto da tekstoj, alireblaj tra la interreto, ebligis evoluigi statistikajn kaj post neŭronajn modelojn.

Certe, bona aŭtomata tradukado baziĝas sur antaŭe bone tradukitaj datenoj, do taŭgaj modeloj dependas ne nur de la arkitekturo de la modelo kaj trejnado, sed ankaŭ de la datenaro kaj metodoj uzataj por certigi la kvaliton dum la evoluigo de tiujn modelojn.

3.1 DLT Distribuita Lingvo-Tradukado

La projekto Distribuita Lingvo-Tradukado (Witkam 1983), gvidata de BSO/Buro voor Systeemontwikkeling en Utrecht (Nederlando), estis ambicia iniciato dum la 1980-aj jaroj celanta krei maŝintraduksistemon, kiu utiligus distribuita komputado.

En 1983 Witkam proponis krei sistemon por duon-aŭtomata tradukado, kie komputilo tradukas de la fontolingvo al pontolingvo, sed ĝi povas ankaŭ konsulti la provizanton de la dokumento. Poste, la tradukado estas farita, en alia loko kaj tempo, tute aŭtomate de la pontolingvo al elektita cellingvo.

Tiu nova koncepto baziĝis sur la uzo de interlingvo sen ambiguecoj por ke la tradukado al iu ajn cellingvo povus esti farita aŭtomate kaj altkvalita. Reguleco de la gramatiko kaj la malpligranda kvanto da plur signifaj vortoj kaj sinonimoj de Esperanto igis ĝin taŭga elekto por ĉi tiu celo.

La fina celo de la DLT-projekto estis subteni multlingvan tradukadon tra pluraj eŭropaj lingvoj, principe de informaj dokumentoj, kio faciligus komunikadon ene de la Eŭropa Ekonomia Komunumo, nun la Eŭropa Unio.

Por forigi ambiguecon DLT uzis modifitan Esperanton, kun signoj por indiki morfemlimojn kaj sintagmofinon. En la kazo de DLT, la projekto celis disvolvi lingvosendependan reprezentadon de signifo, kiu poste povus esti tradukita al ajna cellingvo. Pro la uzo de la pontolingvo, nur du traduk-direktoj ekzistas po lingvo. Tiel, la nombro da eblaj rektaj tradukoj estas reduktitaj kaj la tradukprogramoj estas simpligitaj.

La DLT-projekto estis desegnita por utiligi distribuitan komputadon, kio signifas, ke la tradukprocezo estis dividita inter pluraj komputiloj en malsamaj lokoj, kaj eble en malsamaj tempoj.

DLT komence uzis regulo-bazitan maŝintradukadon (Schubert 1986), kiu inkluzivas disigon de la fontoteksto por analizi ĝian gramatikan strukturon, aplikadon de tradukreguloj, kaj generadon de la celteksto laŭ tiuj reguloj. Tiu aliro postulas intensan laboron, postulante ankaŭ vastan lingvistikan scion kaj rimedojn por kovri ĉiujn eblajn lingvoparojn kaj nuancojn. La uzo de pontolingvo helpas redukti ĉi tiun kompleksecon. En la posta fazo la projekto transiris al perekzempla statistika tradukado.

Tio estis pionira aliro en tiu tempo, konsiderante la limigojn de komputada kapablo kaj malrapidaj retoj dum la 1980-aj jaroj. La DLT-projekto alfrontis multajn teknikajn kaj loĝistikajn defiojn, precipe en la kampo de natura lingvokompreno kaj la limigoj de distribuita komputado dum tiu epoko.

Kvankam la projekto mem ne rezultis en plene funkcia traduksistemo inter pluraj lingvoj, ĝi grave kontribuis al la kampo de maŝintradukado kaj influis estontajn esplorojn kaj evoluojn en ĉi tiu areo.

3.2 SMT Lingvomodeloj

SMT baziĝas sur statistikaj modeloj por antaŭdiri la plej verŝajnan tradukon de teksto surbaze de probablecoj derivitaj el granda datenaro de paralelaj tekstoj, tio estas, fontolingvaj kaj cellingvaj paroj.

La unua paŝo por la kreado de la lingvomodelo estas vicigi la frazojn de la paralela korpuso kaj determini la respondecon inter la vortoj aŭ frazoj en la fontolingvo al iliaj respektivaj tradukoj en la cellingvo.

Post tio, la statistika tradukmodelo estas kreita, ĝi taksas la probablecon de vorto aŭ frazo en la cellingvo, kiu respondas al vorto aŭ frazo en la fontolingvo. La flueco de la cellingvo, generita de la lingvomodelo estas determinita per takso de la probableco de vico da vortoj. Ĉi tio certigas, ke la kreita teksto estas gramatike ĝusta kaj ŝajnas natura. La modelo ankaŭ antaŭdiras kiel la vico da vortoj aŭ frazoj povus ŝanĝiĝi dum tradukado pro konsideroj pri la diferencoj en vortordo inter la fontolingvo kaj cellingvo.

Fine, kiam oni tradukas novan frazon, la SMT-modelo uzas la antaŭe kreitan statistikan modelon por generi aron da eblaj tradukoj. Ĉi tiuj tradukoj estas taksitaj surbaze de iliaj probablecoj derivitaj el la tradukmodelo, lingvomodelo, kaj reordiga modelo. La plej verŝajna traduko, elektita surbaze de la plej alta kombinita probableco, estas liverita kiel la rezulto de la SMT-sistemo.

La kvalito de SMT-rezultoj estas taksita uzante mezuron kiel BLEU, kiu komparas ĉiuj el la maŝinaj generitaj tradukoj al la korpuso de referencaj tradukoj kaj provizas gradon pri la kvalito. Ĉi tiu grado estas uzata por kompari malsamajn modelojn kaj elekti la plej taŭgan por specifa tasko.

Tamen, la efektivigo de SMT-sistemoj fariĝis ebla nur post la kresko de komputila prilabora povo kaj la havebleco de grandaj korpusoj de dulingvaj tekstoj. Inter la defioj de SMT estas la postulo de vastaj kvantoj da datenoj por atingi altan precizecon, pro tio ĝi ofte baraktis kun lingvoj al kiuj mankas ampleksaj paralelaj korpusoj, idiomaj esprimoj, tekstoj ekster la specifa kampo de la datenoj, signifoj kiuj dependas de la kunteksto, kaj dependeco ene de longaj frazoj.

Resume, SMT-modeloj funkcias per eltrovo de statistikaj kongruencoj en granda dulingva datenaro por antaŭdiri la plej verŝajnan tradukon el teksto en la fontolingvo. Ili dependas de kunordigado de tradukmodeloj, lingvomodeloj kaj reordigaj modeloj por generi tradukojn, kiuj estas kaj precizaj kaj fluecaj, tamen ili havas limigojn en prilaborado de kompleksaj lingvaj fenomenoj.

3.3 Lingvaj modeloj kiel nigra skatolo kaj neŭronaj retoj

Se oni rigardas iu ajn el la lingvaj modeloj diskutitaj: RBMT, SMT aŭ NMT kiel nigra skatolo, ili ŝajnas same. Ĉiuj ricevas tekston en la formo de vico da nombroj, ĉar ĉiu lingva signo de la teksto devas esti kodita antaŭ ol ili estas senditaj al la lingva modelo. Ankaŭ la eliro de ĉiuj lingvaj modeloj estas sekvenco da nombroj, kiuj devas esti malkoditaj antaŭ ol ili estas prezentataj al la uzanto de la tradukilo.

Do, oni povas pensi pri tiuj tradukmodeloj kiel matematikaj funkcioj, kiuj ricevas vicon da nombroj, kaj traktas ilin per la interna funkcio de tradukado, kaj prezentas la rezulton kiel alian vicon da nombroj al la uzanto.

Povas esti malfacili malkovri kiun tipon de tradukmodelo ĝi estas se tiu informo ne haveblas. La diferenco estas nur en la maniero kiel ili estis interne kreitaj:

- RBMT-modeloj postulas kreadon de reguloj, kutime per lingvistoj kiuj konas la strukuron de la tradukendaj lingvoj.
- SMT-modeloj postulas malpli da homa laboro por pritrakti la datenojn, ĉar programoj estas uzataj por malkovri la frekvencojn de vortoj, sintagmoj, ktp el la korpusoj de ambaŭ lingvoj, kaj post kunordigi la rilatajn unuojn.

Nun venas du gravaj demandoj:

Ĉu ekzistas ĝenerala komputila modelo, kiu povas esti ŝanĝita por proksimiĝi al tiu matematika funkcio de tradukado?

Ĉu eblas aŭtomate trovi tiun proksimuman matematikan funkcion de tradukado el la korpusoj?

La respondo al ambaŭ estas: Jes!

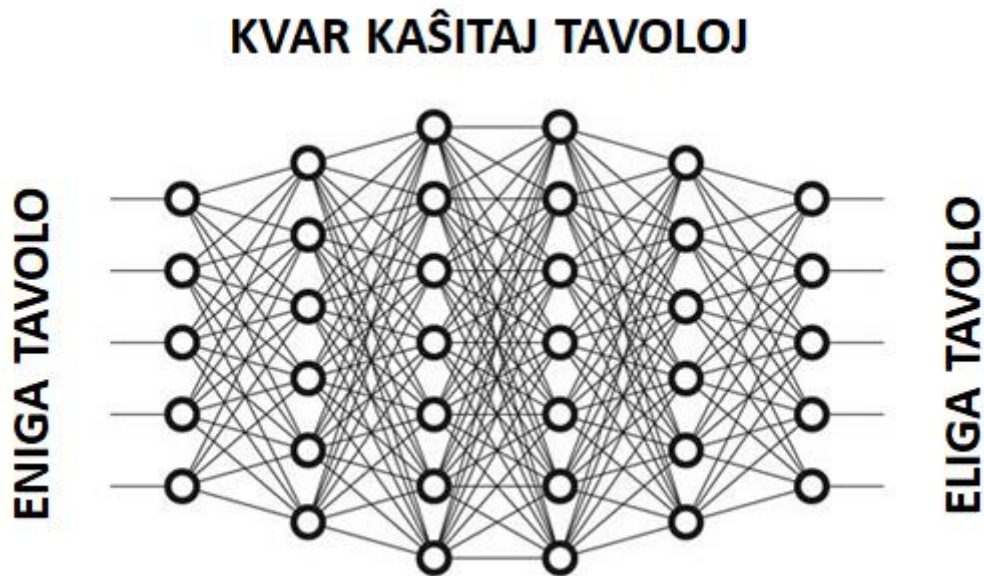
Pluraj artikoloj (Cybenko 1989) (Hornik 1989) (Gripenberg 2003) pruvis ke ekzistas NMT-modeloj, kiuj povas proksimiĝi al iu ajn funkcio simple ŝanĝante iliajn parametrojn. Vere dire, ili ne klarigas kiel eltrovi la parametrojn de tiuj modeloj, ili nur pruvas ke la modeloj ekzistas.

Estas du partoj en la proponitaj modeloj:

- La arkitekturo de la interkonektitaj nodoj;
- La nombraj parametroj alkroĉitaj al la nodoj, ankaŭ nomitaj pezaj.

Tiu proksimuma funkcio estas kreita ŝanĝante la valorojn de la nombraj parametroj ekzistantaj en ĉiu nodo de la modelo.

Kiam nodo ricevas enigon, ĝi uzas siajn parametrojn por generi eligon, kiu estas sendita al pluraj aliaj nodoj. Vidu simplan neŭronan modelon en Bildo 3-1.



Bildo 3-1 Simpla neŭrona modelo

Lernado de la NMT-modelo okazas ŝanĝante la pezojn de la nodoj tiel ke kiam frazo en la fontolingvo estas sendita al la eniga tavolo la eliga tavolo sendas tiun frazon jam tradukita al la cellingvo.

Kaj la arkitekturo de la modelo, kaj la matematika funkcio de la nodoj, kaj kiel trovi la ĝustajn parametrojn uzatajn en la funkcio de tiuj nodoj estas ekster la amplekso de ĉi tiu studo.

3.4 NMT Lingvomodeloj

Same kiel la RBMT-modeloj kaj la SMT-modeloj, la NMT-modeloj ricevas tekston koditan kiel vico da nombroj ĉar la Centra Proceza Unuo (CPU) kapablas fari nur aritmetikajn kaj logikajn operaciojn.

Konputiloj uzataj por trejni modelojn de NMT havas, krom la CPU, ankaŭ Grafikan Procezan Unuon (GPU) aŭ Tensoran Procezan Unuon (TPU).

Konsekvence, artefaritaj intelektaj tekstaj tradukiloj devas kodi literojn, vortojn kaj frazojn kiel vico da nombroj. Por fari tion oni devas unue elekti kiel kodigi tiujn elementojn.

- Literoj estas kutime koditaj en la kodaro Unikodo UTF-8 ĉar ĝi ampleksas ĉiujn homajn lingvojn kaj la unuaj 128 kodoj koincidas kun Askio. Tekstoj skribitaj en la angla kaj aliaj okcidentaj eŭropaj lingvoj, kiuj uzas principe Askio, estas tiel

konservitaj en malgrandaj dosieroj, ĉar Askiaj kodoj estas konservitaj per nur unu bitoko kiam UTF-8 estas uzata.

- Vortoj kaj interpunkcioj en frazoj estas apartigitaj kaj analizitaj en unuoj kiuj havas semantikan enhavon por la modelo. Tiuj unuoj estas aŭtomate eltrovitaj el la korpusoj, tial ili ne respondas al gramatikaj unuoj kiel vortoj, radikoj, afiksoj, ktp. Pro tio oni devas elekti novan nomon por tiuj unuoj. Mi uzas, en tiu ĉi studo, por traduki la terminon “*token*” uzata en la angla la vorton “ĵetono”.

La aro da eblaj ĵetonoj estas la vortprovizo (angle: *vocabulary*) de la tradukmodelo, ili povas esti vortoj, subvortoj, aŭ eĉ unikodaj signoj.

Arbitra nombra kodo estas asignita al ĉiu ĵetono kiam ili estas kreitaj, aldone malsamaj modeloj kutime uzas malsamajn ĵetonojn.

Vortoj, kiuj ne eblas esti konstruitaj per la konataj ĵetonoj, de la vortprovizo, estas ne tradukeblaj.

La NMT-modelo ne rekte uzas la ĵetonojn, ĝi prenas vektoran reprezentadon de la ĵetono de matrico, kiu estis kreita dum la trejnado de la modelo, la rezulto estas vektoro, tio estas: vico da nombroj, kutime enhavante centojn ol milojn da nombroj. Ĉi tiu vektoro enhavas semantikajn informojn pri la ĵetono, kio signifas, ke similaj ĵetonoj (laŭ signifo aŭ kutima uzo) havas vektorojn, kiuj estas proksimaj unu al la alia. En kelkaj NMT-modeloj la vektoroj povas esti pli rafinitaj surbaze de la kunteksto en kiu la ĵetono aperas. Ĉi tio signifas, ke la sama ĵetono povus havi malsamajn vektorojn depende de la aliaj ĉirkaŭaj ĵetonoj (vortoj, interpunkcioj, subvortoj, ktp), permesante al la modelo kapti signifojn, kiuj dependas de la kunteksto.

Tial, la frazoj senditaj al la NMT-modelo estas vico da vektoroj. Ankaŭ la eligo de la modelo estas alia vico da vektoroj, kiu devas esti malkodita por generi la tradukita teksto.

Oni devas noti ke, kvankam la NMT havas grandegan nombron da nodoj, ĝi estas finita kaj havas limigojn, kiel la grandecon de la vortprovizo, la longecon de la frazoj, la kuntekstan fenestron (nombro da ĵetonoj senditaj kune al la NMT), ktp.

Modeloj, en kiuj la reprezentado de vorto dependas de la kunteksto en la frazo, kutime havas fiksan vortprovizon, ofte de 30.000 ĝis 50.000 ĵetonoj, kiu inkluzivas oftajn vortojn, subvortunuojn, kaj ankaŭ unikodaj signoj.

Konklude, NMT-modeloj estas matematikaj funkcioj, kiuj traktas vicojn da nombroj; depende de kiel ili estas organizitaj, ĉi tiuj nombroj povas esti nomataj matrico aŭ tensoro. En tiu ĉi studo ni provas kompreni ĝis kiel la ĵetonoj estas kreitaj, ne zorgante pri kiel ili estas konvertitaj al vektoroj kaj interne uzatajn de la NMT-modelo.

3.5 Uzo de la NMT-modeloj

Nuntempe, tradukiloj estas bazitaj en NMT-modeloj, sed interne ili havas plurajn aliajn modulojn por identigi la lingvon, normaligi la tekston, kodigi la tekston por krei la ĵetonojn, ktp. Oni ne pensas pri tiuj detaloj kiam retaj haveblaj tradukiloj estas uzataj en itereta paĝo.

Programoj ankaŭ simile uzas ilin, sed per aliro al API (Aplika Programara Interfaco) de tradukservilo kiel *Google Cloud Translation API*, *Microsoft Translator Text API*, *Amazon Translate*, *DeepL API*, *IBM Watson Language Translator*, *RWS/SDL Language Cloud*, *SYSTRAN Translate API*, ktp. Tiuj tradukserviloj uzas lingvomodelojn trejnatajn por specifaj traduktaskoj.

Krom ĉi tiuj pagaj nelokdependaj serviloj ekzistas pluraj pretaj senpagaj modeloj uzeblaj ankaŭ surloke, kiam privateco de la dokumentoj gravas.

Facile uzebla babilejo, loke funkcia, kiu, pro tio, konservas loke la privatajn datenojn, kiel [GPT4All](#) ebligas pritraktadon de informoj kaj simplan tradukadon. [Ollama](#)³⁸ estas alia malfermfonta platformo por funkcii, administri kaj konstrui lokajn aplikaĵojn kun grandaj lingvomodeloj.

Ekzistas pluraj ilaroj por disvolviĝo de tradukprogramoj, inter ili, la *Hugging Face Hub*³⁹ estas platformo kun preskaŭ unu miliono da lingvomodeloj, centoj da miloj da datenaroj kaj centoj da miloj da demostraj aplikaj programoj nomitaj Spacoj (angle: *Spaces*), publike haveblaj, en reta platformo kie homoj povas facile kunlabori kaj kune konstrui maŝinan lernadon.

Kelkaj modeloj estas trejnataj uzante datenaron, kiu enhavas centojn da lingvoj, aliaj estas fajne trejnataj por tradukado en specifa paro da lingvoj. Vidu kelkajn ekzemplojn en **Tabelo 3-1**.

MODELO	LINGVOJ
facebook/nllb-200-distilled-600M	196
Unbabel/TowerInstruct-7B-v0.2	10
google-t5/t5-base	4 - Angla, franca, rumana, germana
google/bert2bert L-24 wmt de en	2 - De la germana al la angla
Helsinki-NLP/opus-mt-en-eo	2 - De la angla al la esperanta

Tabelo 3-1 Ekzemploj de lingvaj modeloj uzataj por tradukado.

Por la trejnado kaj fajna trejnado de la modeloj oni uzas ĝeneralaj aŭ specifaj datenaroj. Vidu en **Tabelo 3-2**.

³⁸ <https://ollama.com>

³⁹ <https://huggingface.co/models>

DATENARO	LINGVOJ
Muennighoff/flores200	200
Helsinki-NLP/opus-100	100 - Angla-centra plurlingva korpuso
aiana94/polynews-parallel	64 - Dateno enhavanta novaĵtitolojn por 833 lingvoparoj
Helsinki-NLP/europarl	21 - Paralela korpuso el la retejo de la Eŭropa Parlamento

Tabelo 3-2 Ekzemploj de datenoj uzataj por tradukado.

Por kompari lingvajn modelojn oni uzas anglajn kaj tradukitajn datenojn, kiel la anglajn COPA kaj [StoryCloze](#)-datenojn kaj ĝiajn tradukojn al 10 ne anglaj lingvoj: [XCOPA](#) kaj XStoryCloze. Vidu ofte uzatajn datenojn en la **Tabelo 3-3**.

DATENARO	LINGVOJ
LSDSem/story_cloze	1 – Angla
juletxara/xstory_cloze	11
pkavumba/balanced-copa	1 – Angla
XCOPA	11
EleutherAI/lambada_openai	5 – Angla, germana, hispana franca, itala
Muennighoff/xwinograd	6 – Angla, franca, japana, portugala, rusa, china

Tabelo 3-3 Ekzemploj de datenoj por kompari lingvajn modelojn.

Resume, ekzistas grandega nombro da lingvaj modeloj kiuj povas esti uzataj de homoj rekte en interretaj paĝoj ol per programoj, ili ankaŭ povas esti uzataj surloke de homoj kaj per programoj. Sed pluraj faktoroj devas esti konsiderataj por elekti la plej bonan tradukmodelon por specifa uzo. Aldone, la lingva modelo provizas nur la unuan tradukadon, kiu devas esti kontrolita, korektita kaj aprobita. Tial, eĉ uzante bonan lingvan modelon la tradukado postulas ankaŭ aliron al memoro de tradukado, terminaro kaj tradukistoj.

4 Kodiĝo de la datenoj

Tekstoj devas esti koditaj antaŭ ol ili povas esti senditaj al la lingvaj modeloj. Tri etapoj estas analizataj en tiu ĉi ĉapitro, la kodiĝo de la signoj per la unikoda kodaro, la normaligo de la teksto por malgrandigi la kvanton de la uzataj kodoj kaj la ĵetonigado por ke la teksto povu esti uzata en specifa tradukmodelo.

4.1 La Unikoda kodaro

Nuntempe, la grupo da bitoj uzata en telekomunikado, kiun komputiloj ofte traktas kiel unuo estas la okbita bajto nomata bitoko. Unu bitoko enhavas 8 bitojn, do ĝi povas reprezenti 256 nombrojn de 0 ĝis 255, kutime scribitaj en la deksumo bazo, per la 16 sekvantaj ciferoj: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A,B ,C, D, E, F, de %00 ĝis %FF. La % antaŭ la deksumo nombro estas la norma kodiĝo per elcentosigno uzata en interretaj adresoj por sendi ne Askian aŭ Askian ne videblajn kodojn ([IETF RFC3986](https://datatracker.ietf.org/doc/rfc3986/))⁴⁰. En XML dokumentoj oni kodiĝas tiuj nombroj alie, jene: de � ĝis ÿ aŭ de � ĝis ÿ.

Por reprezenti ĉiujn eblajn signojn (literojn, ciferojn, interpunkciojn, piktogramojn, stirsignojn, ktp) oni devas uzi pli ol unu bitokon. Pro tio multaj kodoj estis inventitaj, el kiuj, du gravaj estas Askio kaj Unikodo.

Askio enhavas nur 128 bitokojn de %00 ĝis %7F. Ĝi estas uzata por anglaj tekstoj kaj por skribi programojn.

Unikodo havas kodopunktojn por ĉiuj homaj lingvoj. Kiam Unikodo estas uzata en interreta adreso la kodopunkto estas kodata per elcentosignoj en UTF-8 Unikoda Transforma Aranĝo (angle *Unicode Transformation Format*) laŭ [IETF RFC3629](https://datatracker.ietf.org/doc/rfc3629/)⁴¹. Vidu ekzemplon de tiu kodiĝon en Tabelo 4-1.

GAMO DA KODOPUNKTOJ (deksumo)	UTF-8 BITOKA SEKVENCO (duuma)	UTF-8 BITOKOJ
0000 0000 ĝis 0000 007F	0xxxxxxxx	1
0000 0080 ĝis 0000 07FF	110xxxxx 10xxxxxx	2
0000 0800 ĝis 0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx	3
0001 0000 ĝis 0010 FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx	4

Tabelo 4-1 Kodopunktoj en la UTF-8 Unikoda Transforma Aranĝo.

Ekzemple, la litero “ĝ” povas esti prezentata kiel deksumo kodo, duuma kodo, elcenta kodo kaj aliaj. Vidu tiujn formojn en Tabelo 4-2.

LITERO	DEKSUMO KODO	UTF-8 DUUMA KODO	% KODIĜO EN UTF-8
ĝ	\u011D	11000100 10011101	%C4%9D

Tabelo 4-2 Pluraj kodoj por reprezenti la literon ĝ

⁴⁰ <https://datatracker.ietf.org/doc/rfc3986/>

⁴¹ <https://datatracker.ietf.org/doc/rfc3629/>

La eskapa sekvenco `\u` plus 4 deksumaj ciferoj estas uzata en programoj por reprezenti kodojn de la baza plurlingva ebena de Unikodo. Por la aliaj ebenaĵoj oni uzas du `\u` sekvencojn.

Oni vidas, en Tabelo 4-1, ke per UTF-8 la unuaj 128 kodopunktoj estas reprezentataj per unu bitoko, kaj tial koincidas kun Askio.

Ekde la invento de la telegrafio, multaj kodoj estis kreitaj por komunikado. La plej simplaj kodaroj enhavas nur kodojn por literoj kaj ciferoj, post interpunkcioj estis aldonitaj. Kune kun la invento de la telekso, por regi la teletajpilon, ankoraŭ aliaj kodoj estis bezonataj.

Unu el la plej gravaj kodaroj, ĝis hodiaŭ uzata, estas la Askia 7-bita signaro (*ASCII American Standard Code for Information Interchange*), trovebla en la unua interreta paĝo de Unikodo: <https://www.unicode.org/charts/PDF/U0000.pdf>. Ĝi enhavas 96 aperigeblajn signojn (literojn, ciferojn, interpunkciojn, ktp.) kaj 32 stirsignojn por regi la komunikmedion.

En C, JAVA, ktp programoj, por uzi la ne aperigeblajn stirsignojn oni povas uzi la deksumajn ciferojn aŭ, por kelkaj ofte uzataj stirsignoj, specifa litero antaŭita per la eskapa signo `\`. Vidu ekzemplojn en Tabelo 4-3.

DEKSESUMAJ CIFEROJ	ESKAPA SEKVENCO	UZO
<code>\u0008</code>	<code>\b</code>	Retropaŝo (angle: <i>BS BackSpace</i>)
<code>\u0009</code>	<code>\t</code>	Tabelsalto (angle: <i>HT Horizontal Tab</i>)
<code>\u000A</code>	<code>\n</code>	Linifino en Unikso (angle: <i>LF Line Feed</i>)
<code>\u000C</code>	<code>\f</code>	Nova paĝo (angle: <i>FF Form Feed</i>)
<code>\u000D</code>	<code>\r</code>	Linifino en Mac OS (<i>CR Carriage Return</i>)
<code>\u0022</code>	<code>\"</code>	Citilo
<code>\u002F</code>	<code>\ /</code>	Suprenstreko
<code>\u005C</code>	<code>\\</code>	Malsuprenstreko

Tabelo 4-3 Kodiĝo en C kaj JAVA por ne aperigeblaj stirsignoj.

En Vindozo, vidu en Tabelo 4-4, malsimile ol en Unikso kaj Mac OS, du stirsignoj, kune, estas uzataj por indiki linifinon.

DEKSESUMAJ CIFEROJ	ESKAPA SEKVENCO	UZO
<code>\u000D \u000A</code>	<code>\r \n</code>	Linifino en Vindozo

Tabelo 4-4 Linifino en Vindozo.

Por ebligi ke ununura kodaro estu uzata por ĉiuj homaj lingvoj la universala kodaro Unikodo estis kreita. Ĝiaj 128 unuaj kodopunktoj estas la Askia signaro. Unikodo enhavas kodopunktojn ne nur por literoj, ciferoj, interpunkcioj, diakritoj, ktp sed ankaŭ por piktogramoj kiel ☺, 🌐, 🚗. Montritaj ĉi tie uzante la tiparon *Segoe UI Symbol*.

Tamen, Unikodo ne difinas la signotiparon, pro tio, se la ĝusta tiparo ne estas elektita aŭ ne ekzistas en la operaciuma sistemo, multaj kodopunktoj ne estas videblaj. Ekzemple, la tri supraj piktogramoj, nun en la tiparo *Times New Roman* videblas tiel: □, □, □.

Ankaŭ, Unikodo ne difinas kiel la kodopunktoj estas koditaj en programoj kaj operaciumaj sistemoj. Pro tio, kiam oni konservas dosieron kiel *.txt per Notepad.exe en Vindozo oni povas elekti kodiĝon kiel:

ANSI (Askio plus aliaj 128 kodoj)

UTF-8 (Unikodo kodita per bitokaj unuoj)

UTF-16 (Unikodo kodita per deksumaj unuoj)

Eblas ankaŭ aldoni, pere de 2 aŭ 3 bitokoj, la Unikodon \uFEFF nomata BOM (angle: *Byte Order Mark* - Bajta Orda Marko), en la komenco de la dosiero por indiki la tipon de kodiĝo. Vidu tiujn markojn en Tabelo 4-5. Por ke malnovaj programoj, kreitaj antaŭ la ekesto de Unikodo, povu uzi dosierojn konservitajn en UTF-8, kiuj enhavas nur Askion, oni kutime ne aldonas la nedevigan BOM al UTF-8 dosieroj.

TIPO DE KODIĜO DE LA BOM	KOMENCAJ BITOKOJ EN LA DOSIERO
BOM en UTF-8	EF BB BF
BOM en UTF-16 BE (angle: <i>Big Endian</i>)	FF FE
BOM en UTF-16 LE (angle: <i>Little Endian</i>)	FE FF

Tabelo 4-5 Ebla, nedeviga, Bajta Orda Marko en la komenco de dosiero.

La Unikodaj kodopunktoj estas difinitaj per deksumaj ciferoj en 17 plurlingvaj ebenaĵoj (angle *Multilingual Planes*). Vidu ilin en Tabelo 4-6.

DEKSESUMAJ CIFEROJ	UZO
\u0000 \u0000 ĝis \u0000 \uFFFF	0 - Baza plurlingva ebenaĵo
\u0001 \u0000 ĝis \u0001 \uFFFF	1 - Suplementa plurlingva ebenaĵo
\u0002 \u0000 ĝis \u0002 \uFFFF	2 - Suplementa piktograma ebenaĵo
\u0003 \u0000 ĝis \u0003 \uFFFF	3 - Terciara piktograma ebenaĵo
\u0004 \u0000 ĝis \u000D \uFFFF	4 ĝis 14 - Ne asignitaj ebenaĵoj
\u000E \u0000 ĝis \u000E \uFFFF	14 - Suplementa speciala cela ebenaĵo
\u000F \u0000 ĝis \u0010 \uFFFF	15 kaj 16 - Suplementa privata ebenaĵo

Tabelo 4-6 La 17 plurlingvaj ebenaĵoj de Unikodo.

Krom UTF-8 kaj UTF-16 ekzistas aliaj kodiĝoj por la Unikodaj kodopunktoj, kiel UTF-32, UTS kaj UTF, por detaloj vidu https://www.unicode.org/faq/utf_bom.html.

Unikodo provas krei ununuran kodaron por ĉiuj homaj lingvoj, tamen la diversaj uzoj de la samaj simboloj faritaj de diversaj landoj, kulturoj, fakoj, ktp kreas malfacilaĵojn. Unue, Unikodo difinas la kodopunktojn, ne la tiparon de tiuj kodoj. Pro tio la sama kodopunktoj kutime havas plurajn simbolojn dependante de la uzata tiparo. Vidu kelkajn el ili en Tabelo 4-7.

SIMBOLO LAŬ LA TIPARO	KODOPUNKTO	TIPARO
<i>a</i>	\u0041	Lucida Handwriting
ⴰ	\u0041	Algerian
a	\u0041	Courier New
a	\u0041	Calibri
a	\u0041	Times New Roman

Tabelo 4-7 Sama kodopunkto havas plurajn simbolojn laŭ la tiparo.

Ankaŭ, malsamaj kodopunktoj povas uzi saman simbolon.

Ekzemple, kelkaj simboloj en la Tabelo 4-8 ŝajnas same, sed havas malsamajn kodopunktojn.

SIMBOLO	KODOPUNKTO	UZO
A	\u0041	Latina litero
A	\u0410	Cirila litero
A	\u0391	Greka litero
C	\u0043	Latina litero
C	\u216D	Roma numeralo
ñ	\u0043	Hispana litero
ñ	\u006E (n) kaj \u0303 (kombinanta tilde ~)	Hispana litero

Tabelo 4-8 Pluraj kodopunktoj povas havi la saman simbolon.

Por limigi la vastecon de la vortprovizo, ne ĉiuj Unikodaj kodopunktoj estas utiligitaj en la NMT. Antaŭ tradukado la teksto estas normaligita, tiel ke inter aliaj simpligoj, ortografiaj varioj estas sisteme simpligitaj kaj malpli da kodopunktoj estas uzataj.

4.2 Normaligo de la teksto

Estas pluraj eblaj aplikoj de normaligo; unu el ili estas certigi, ke tekstaj datenoj estas preparitaj tiel, ke ili faciligas efikan lernadon kaj tradukadon en NMT. Normaligo estas ankaŭ uzata en la studo kaj analizo de historiaj lingvaj tekstoj (Bawden 2022) kaj (Tjong Kim Sang 2017), kie normigado de ortografio estas utila, ĉu la analizo estas farita per fakuloj aŭ per aŭtomata analizo uzante ilojn por Natura Lingva Prilaborado (NLP). Eisenstein 2013 diskutas kelkajn el la problemoj de teksta normaligo: ne ĉiam estas klare, kiu normo devus esti uzata por normaligi, kaj la normaliga paŝo povas ŝanĝi la signifon de teksto.

Malgraŭ tio, pro praktikaj limigoj, pluraj normaligaj proceduroj estas ofte uzataj, ĉar ĝi ne nur helpas homogenigi variablajn ortografiojn, numerajn formatojn, interpunkciojn, sed ankaŭ plibonigas stabilecon de la trejnado kaj pli koheran uzon de NMT per malgrandigo de la vortprovizo kaj de la komplekseco de la MT modelo.

Kelkaj el la eblaj normaligaj proceduroj estas:

- **Ŝanĝi usklecon de la teksto:** Konverti ĉiujn signojn en la teksto al minusklaj aŭ majuskulaj signoj, kiel postulas la tradukmodelo, por certigi unuformecon;

- **Forigi aŭ normaligi interpunkciojn:** Unikodo havas 278 tipojn da interpunkcioj, 30 tipojn da strekoj, 11 tipojn da ligiloj, ktp⁴²;
- **Normaligi ortografion:** Sistemigi ortografiajn variantojn, kiel la uzo de Esceto (ß) en Germanio kaj Aŭstrio kontraŭ duobla-s en Svislando kaj Liĥtenŝtejno;
- **Anstataŭigi inter-vortajn spacetojn:** Anstataŭigi nerompeblajn spacetojn aŭ nedeziratajn signojn de la teksto, kiel kromajn spacetojn, tabelsaltojn aŭ linifinojn per normalaj spacetoj (Unikodo havas pli ol 20 tipojn da spacetoj);
- **Malgrandigi la nombron de uzataj citiloj:** (Unikodo havas ĉirkaŭ 30 tipojn da citiloj);
- **Anstataŭigi kombinantajn signojn per antaŭe kombinitaj signoj:**
Ekzemple: ñ (n+~) per ñ.

NMT-modeloj havas limigojn en la grandeco de la vortprovizo kaj en la nombro da konataj signoj, limigi la kompleksecon de la teksto estas artifiko por atingi pli precizan tradukon de la vortoj, eĉ se la tekstoformatado estas pli simpla kaj nuancoj de signifoj povas esti perditaj.

4.3 Ĵetonigo de la teksto

Pluraj provoj estis faritaj por trakti tekstojn kaj apartigi semantikajn unuojn. La unuaj, puraj proceduraj metodoj, per fiksataj reguloj, funkcias por specifaj tekstoj, sed nur en la lingvo kaj la kunteksto por kiuj ili estis planitaj; la ekesto de paralelaj komputiloj, kun granda kaj rapida memoro, lerteco por matrica kalkulado kaj grandaj korpusoj ebligas serĉado por aliaj empirikaj metodoj kiuj povas esti aplikataj en diversaj malsamaj kunteksto.

Ekzemple, la [NLTK Natura Lingva ilaro \(angle: Natural Language Toolkit\)](#)⁴³ estas platformo por konstrui *Python*-programojn por prilabori homajn lingvajn datenarojn. Ĝi provizas bibliotekojn por klasifiko, ĵetonigo, derivado, etikedado, analizado kaj semantika rezonado de tekstoj. Tiu biblioteko provizas plurajn ĵetonigilojn, inter ili oni trovas la procedurojn en Tabelo 4-9, eblas ankaŭ specifi, al la proceduroj, la lingvon kaj aliajn specialajn trajtojn per parametroj.

PROCEDURO	ĴETONIGA FUNKCIO
<code>split(frazo)</code>	apartigas vortojn ĉe spacetoj
<code>word_tokenize(frazo)</code>	apartigas vortojn kaj interpunkcioj
<code>wordpunct_tokenize(frazo)</code>	apartigas vortojn, interpunkciojn per regulaj esprimoj
<code>sent_tokenize(frazoj)</code>	apartigas frazojn
<code>TweetTokenizer().tokenize(frazo)</code>	apartigas frazojn de tujmesaĝiloj
<code>ToktokTokenizer().tokenize(frazo)</code>	por angla, persa, rusa, ĉeĥa, franca, germana, vjetnama

Tabelo 4-9 Proceduroj por ĵetonigo en la NLTK.

Tiuj tradiciaj ĵetonigiloj estas planitaj por specifaj uzoj kaj ĉiu el ili provizas rezultojn por aparta fazo de la ĵetonigado, inter aliaj per: apartigo de frazoj, apartigo de vortoj, uzo de

⁴² <https://www.unicode.org/Public/UNIDATA/PropList.txt>

⁴³ <https://www.nltk.org/>

regulaj esprimoj, kapablo por pluraj lingvoj, ebleco rekrei, el la ĵetonoj, la originan frazon, ebleco registri la pozicion de la vortoj en la origina frazo, ktp.

Ĵetonigo estis vaste rigardata kiel solvita problemo pro la alta precizeco de la regulbazitaj ĵetonigiloj, sed regulbazitaj ĵetonigiloj estas malfacile adaptigitaj al novaj kampoj de uzo kaj iliaj reguloj aplikeblas nur al specifa lingvo.

Tiuj proceduroj uzas regulajn esprimojn, statistikajn metodojn kaj povas funkcii kiel parto de la ĵetonigado, sed ne limigas la vastecon de la vortprovizo. Tial aliaj ĵetonigiloj estis proponitaj kaj sukcese estas uzataj en NMT de Google, Microsoft, Meta, ktp.

- *BPE*: Kodado de Bajtaj Paroj (angle: *Byte Pair Encoding*), bazita surbase de kunprema algoritmo (Gage 1994), estas uzata por solvi la problemon de tradukado de raraj vortoj en la ĵetonigilo de lingvaj modeloj kiel [GPT-2⁴⁴](#) (Sennrich 2016a) kaj [GPT-3⁴⁵](#) (Brown 2020). Ĝi povas esti plibonigita per uzo de unulingva datenaroj (Sennrich 2016b).

BPE trovas la plej oftajn parojn da kodopunktoj aŭ subvortoj en la datenaro, identigas ilin kiel nova ĵetono kaj aldonas ilin al la vortprovizo. Post rekuras plurajn fojojn la datenaro, nun konsiderante la novajn ĵetonojn, kaj ripetas la algoritmon por kunpremi la datenaron. La plej oftaj vortoj finiĝas kun ĵetonaj identigiloj al si mem, sed la pli maloftaj vortoj fine konsistas el vortpecoj. BPE ankaŭ aldonas ĉiujn kodopunktojn, kiuj aperas en la datenaro, uzata dum la trejnado, kiel ĵetonojn en la vortprovizo por enkalkuli pri la ebla okazaĵo de maloftaj nekonataj vortoj dum la uzo de la NMT.

- *SentencePiece*: (Kudo 2018b). Ĉi tiu ĵetonigilo estas uzata en la sekvantaj lingvaj modeloj: ALBERT (Lan 2020), T5 (Raffel 2020), XLNet (Yang 2020).
- *Unigram*: En la *Unigram* (Kudo, 2018a) algoritmo la vortprovizo estas konstruita surbase de la ofteco de subvortaj unuoj en la korpuso uzata dum la trejnado.
- *WordPiece*: (Schuster 2012). Ĉi tiu ĵetonigilo estas uzata en la sekvantaj lingvaj modeloj: BERT (Devlin 2019), ERNIE (Sun 2019). Ĝi estas simila al BPE sed uzas malsaman kunfandan strategion. WordPiece celas ekvilibrigi la vortprovizon kaj la nombron da subvortaj unuoj.

Nur vortoj, kiuj apartenas al la vortprovizo de la NMT povas esti tradukitaj, pro tio, pluraj procedoj estas efektivigitaj per ĵetonigilo por akiri la ĵetonojn antaŭ ili estas konvertitaj al vektoroj kaj senditaj al la NMT. Por tiuj vortoj nekonataj aŭ maloftaj, kiuj ne apartenas al la vortprovizo, kaj ne povas esti koditaj per subvortaj unuoj, oni devas anstataŭigi ilin per speciala ĵetono, ekzemple [UNK] (Ne Konata, angle: *UNKnown*).

Ĉiu NMT havas sian propran ĵetonigilon, kiu konsistas el kelkaj el la jenaj partoj:

⁴⁴ <https://github.com/openai/gpt-2>

⁴⁵ <https://github.com/openai/gpt-3>

- Normaligo: baza kaj unikoda normaligo;
- Antaŭtraktado: disigo de ĵetonoj kaj prilaboro;
- Ĵetoniga modelo: ĵetoniga algoritmo - BPE, vortbazita, kodopunkto bazita, ktp;
- Posttraktado: detranĉado, kompletigo, masko.

Kiam oni uzas pretan lingvan modelon, gravas ke la ĵetonigilo, uzata por sendi novajn tekstojn al la modelo, estas la sama uzata por la origina trejnado de la lingva modelo.

Ekzistas bibliotekoj en altnivelaj programlingvoj, kiel *Python*, uzataj por ĵetonigado kaj interfacoj kun la lingvaj modeloj.

Ekzemple, la biblioteko [Tokenizers](https://pypi.org/project/tokenizers/)⁴⁶ de Hugging Face havas la sekvajn ĉefajn trajtojn:

- Ebligas trejnado de nova vortprovizo kaj ĵetonigado uzante la metodojn *Bert* *WordPiece* kaj 3 versioj de BPE;
- Ege rapida por trejnado kaj ĵetonigado, ĉar ĝi estas efektivigita en la Rust programlingvo;
- Dezajnita por esplorado kaj produktado;
- Normaligo estas paralele vicigita kun la originala teksto por ebligi retrovon de la parto de la frazo, kiu respondas al la ĵetono;
- Enhavas la traktadon de detranĉado, kompletigo kaj aldono de bezonataj specialaj ĵetonoj.

Informoj pri la detaloj de la ĵetoniga traktado estas konservita kune kun la NMT en dosiero nomita `tokenizer.json`. Ĉi tiu dosiero uzas la interŝanĝan formaton JSON por dokumenti la parametrojn por la ĵetonigado, vidu en Tabelo 4-10 kelkaj el la parametroj de la NMT `bert-base-cased`, kiu troveblas ĉe la interreta paĝo <https://huggingface.co/google-bert/bert-base-cased/tree/main>.

PARAMETRO	VALORO
<i>version</i>	1.0
<i>truncation</i>	<i>null</i>
<i>padding</i>	<i>null</i>
<i>added_tokens</i>	Listo enhavanta 5 ĵetonojn: [PAD], [UNK], [CLS], [SEP] kaj [MASK]
<i>normalizer</i>	"type": " <i>BertNormalizer</i> "
<i>pre_tokenizer</i>	"type": " <i>BertNormalizer</i> ",
<i>post_processor</i>	"type": " <i>TemplateProcessing</i> "
<i>decoder</i>	"type": " <i>WordPiece</i> "
<i>model</i>	"type": " <i>WordPiece</i> " kaj listo enhavanta la 28996 vortprovizajn ĵetonojn

Tabelo 4-10 Parametroj por la ĵetonigado de la NMT modelo `bert-base-cased`.

⁴⁶ <https://pypi.org/project/tokenizers/>

Kodiĝo ŝanĝas la tekston kaj povas forigi gravajn partojn kiam kodaro, kiu havas malpli da eblecoj estas uzata. En maŝina tradukado la etapo de normaligado simpligas la tekston, pro tio ĝi povas forigi nuancojn de signifo, kaj malhelpi la tradukadon.

5 Uzoj de Neŭronaj Maŝinaj Tradukmodeloj

Maŝina tradukado rapide kaj facile transdonas informojn de unu al alia lingvo, kvankam la kvalito estas konstante kritikata pro manko de flueco, bazaj eraroj kaj ebleco de miskompreno. Tamen novaj tradukiloj neatendite estas haveblaj en saĝaj telefonoj, komunikiloj kaj retejoj de pluraj entreprenoj.

Praktika uzo de tiu ebleco okazis lastatempe en reta kunveno kun kolego el Barato; kiam li kunhavigis sian ekranon, por diskuti kelkajn problemojn, mi vidis ke li uzis ilon por traduki la ricevitan paroladon. Verŝajne mia angla parolado ne estas sufiĉe bona.

Aliflanke, mia poŝtelefono havas ilon en *WhatsApp* por transkribi mesaĝojn ricevitaĵoj el la sekvaj lingvoj: angla, franca, germana, hindia, hispana, itala, kaj japana. Mi jam povis kontroli la kvaliton de la transkribo de ĉiu taga parolado el la angla, hispana kaj japana, kaj mi konstatis ke ili estas tre bonaj. Ankoraŭ ne estas ebleco por kopii kaj traduki, tamen ili helpas min ĉar mi povas tajpi la mesaĝon en tradukilo.

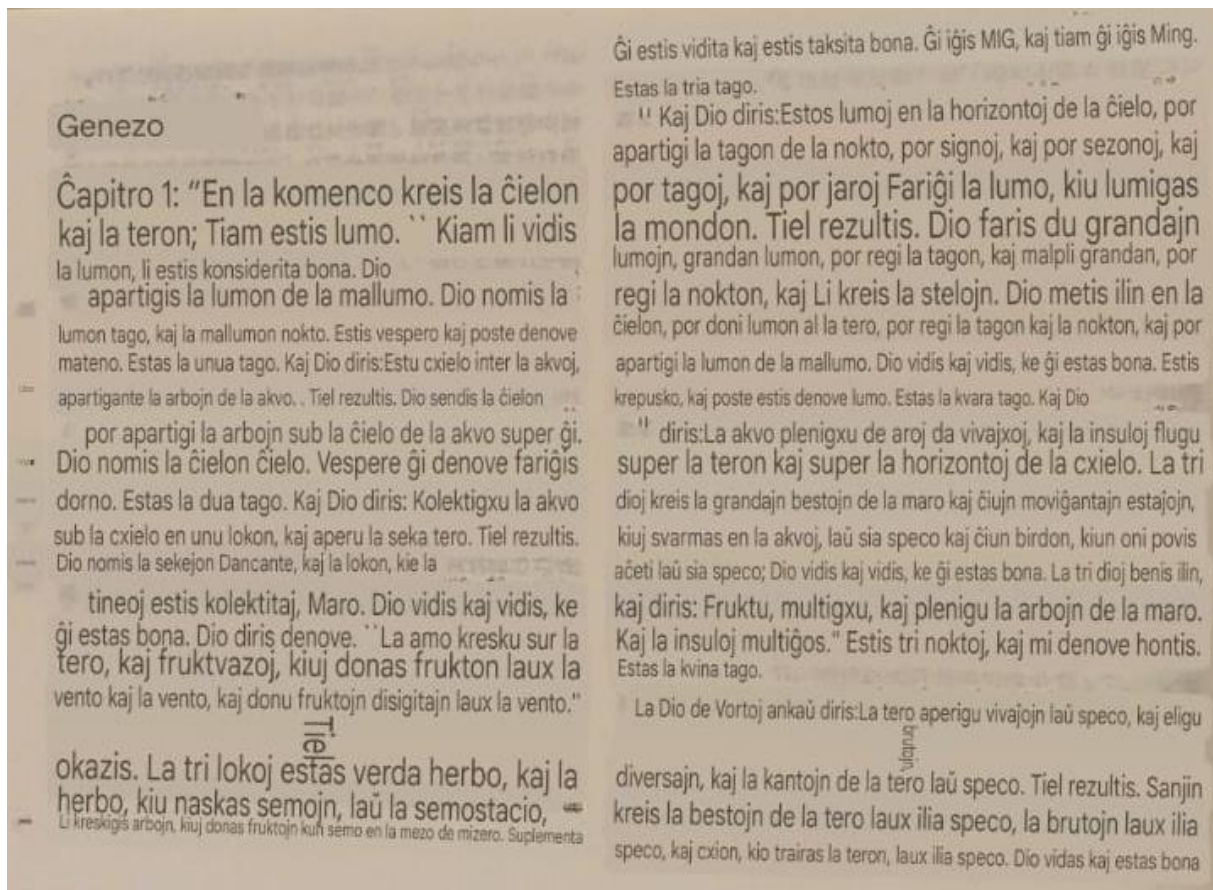
En mia laboro, dum kelkaj mondaj kunvenoj, la transkribita teksto de la prezentado estas tradukita el la angla al kelkaj lingvoj, kiel al la franca, germana, kaj hispana. Nun mi rememoras similan prezentadon, kiu okazis antaŭ dek aŭ dek kvin jaroj, en kiu profesiaj interpretistoj partoprenis, la kvalito ne estis tiel bona, ĉar ili ne konis la teknikajn terminojn kaj ne havis antaŭan aliron al la enhavo de la prezentado. Hodiaŭ jam ekzistas teknologio por helpi interpretistojn, kvankam ĝi ankoraŭ bezonas plibonigojn, kiel estas diskutita, en la artikolo "*Language Technology for Interpreters: The Vip Project*" (Pastor 2020).

Certe, estas pluraj mankoj en maŝina tradukado kaj ĝi ne estas perfekta, sed plurfoje oni bezonas nur kompreni la esenco de la origina mesaĝo, eĉ en mia propra lingvo, dum ĉiutaga konversacio estas pluraj breĉoj kaj nekomprenejoj, tamen la rapido estas pli grava ol la ĝusta gramatiko kaj precizeco.

Retumiloj povas esti agorditaj por aŭtomate traduki la enhavon de interretaj paĝoj, tiel ebligante la uzon de paĝoj skribitaj en nekonataj lingvoj.

Videoj de *Youtube* estas aŭtomate subtitolitaj per maŝina tradukado ebligante iu ajn, kiu scipovas legi, kompreni la videojn, dokumentajn filmojn, kursojn, instrukciojn, ktp.

Alia ebleco, kiun mi plurfoje uzis, estas foti tekston, ne gravas se ĝi estas mane skribita aŭ tajpita, kaj tuj traduki per la aplikaĵo *Translate* de *Google*. Ekzemple vidu la tradukon de la komenco de la biblio el la japana en la Bildo 5-1. Tiu aplikaĵo ebligas ankaŭ kundividi la foton aŭ audi la tekston.



Bildo 5-1 Traduko de foto de la biblio el la japana per *Google Translate*.

Entute, artefarita intelekto kaj maŝina tradukado jam estas parto de nia vivo, la novaj generacioj vivas ekde infaneco en interkonektita mondo, kiu disponigas plurajn novajn eblecojn, defiojn kaj minacojn. Kontinua lernado kaj esplorado estas parto de la nuntempaj mondaj trajtoj.

6 Kiel malaltigi la tradukkostojn

En mia laboro, pro la ofta ŝanĝoj en la teknikaj dokumentoj kaj konsekvencaj bezonoj de novaj tradukoj, oni provas trovi manierojn malaltigi la kostojn de la eksteraj tradukagentejoj.

Mallongaj esploraj projektoj por pruvo de koncepto estis faritaj por taksii la uzon de novaj teknologioj. Tiuj esploroj montris ke ekzistas kelkajn eblecojn pro la facila reta aliro kaj pluraj novaj NMT-modeloj. Du opcioj estis testitaj:

- Prototipo de loka (ene de la interna reto de la entrepreno) maŝina tradukado;
- Uzo de interreta platformo por tradukado.

6.1 Loka maŝina tradukado

Unue la formato de la ekzistantaj dokumentoj estis analizitaj por trovi kiel ĉerpi la tradukendajn segmentojn el la fontodokumentoj kaj kiel uzi la metadatenojn por decidi kiel prilabori tiujn segmentojn.

Poste la memoro de tradukado kaj la terminaro estis konservitaj en vektora datenbanko por ke ili povu esti alireblaj per simileco de vektoroj, ne nur por serĉo de vortoj per regulaj esprimoj. Tiuj vektoroj estas same kiel la vektoroj senditaj al la tradukmodelo.

Fine, la uzebleco, ne la kvalito, de unu NMT-modelo estis taksita.

La konkludo estas ke ekzistas tro granda nombro da similaj NMT-modeloj, kaj ili konstante ŝanĝiĝas. Eblas uzi ilin, sed la dokumentado ne estas bona, la komputiloj devas esti prilabori pova kun memoro de dekoj da gigabajtoj kaj aliro al la GPU.

Oni decidis ke tio estas tro granda klopodo, kaj ne povas esti finance pravigita, pro tio ke ĝi estas nur flankaj projektoj de la entrepreno, ne por krei novan komerceblan produkton.

6.2 Interreta platformo

Ekzistas pluraj pagaj kaj senpagaj platformoj por traduki dokumentojn. Unu el ili, nomata [SmartCat](https://www.smartcat.com/)⁴⁷, provizas kaj tradukilon, kaj aliron al NMT-modeloj, kaj la eblecon dungi tradukistojn el la tuta mondo.

Ĝi estas nur platformo, kiu vendas aliron al maŝina tradukado, kaj laboron de tradukistoj tra programo por maŝina tradukado. SmartCat memorigas min pri *Airbnb*, *Uber Eats*, kaj aliaj platformoj. Mi pensas ke tiu tipo de servo povas iel ŝanĝi kiel tradukistoj laboras, ĉar ĝia kosto baziĝas sur la nombro da tradukitaj vortoj. La propagando deklaras averaĝe US\$ 0.04-US\$ 0.06 po vorto kontraŭ US\$ 0.16 en aliaj agentejoj. Mi uzis, per la platformo, enigon de dosieroj en la formatoj dita, ditamap kaj xcliff; ankaŭ aŭtomata tradukado per la NMT-modeloj de ModernMT kaj Google estis uzata en la platformo.

⁴⁷ <https://www.smartcat.com/>

Avantaĝoj:

- Aliro al pluraj pagaj komercaj tradukiloj:
Google, Amazon, DeepL, ModernMT, Microsoft, Yandex, Baidu, kaj OpenAI GPT;
- Ebleco uzi tradukmemoron, kaj terminaron;
- Enigo de pli ol 50 malsamaj formatoj de dosieroj:
docx, doc, txt, rtf, odt, md, mkd, mdwn, mdown, mdtxt, mdtext, markdown, pptx, ppsx, potx, ppt, pps, odp, pdf, xls, xlsx, xlsx, xlsx, xlsx, html, htm, php, bmp, pcx, dcx, png, jp2, jb2, jpc, jpg, jpeg, jfif, tif, tiff, gif, pot, zip, srt, vtt, idml, inx, djvu, djv, dcx, pcx, xml, inc, dita, ditamap, xlf, xliff, mxliff, yml, yaml, json, tjson, locjson, resx, po, properties, strings, mif, mp4, mpeg, avi, mov, 3gp, 3g2, flv, m2v, m4v, mkv, mpg, ogv, qt, vob, wmv, mp3, wav, wma, mp2, ogg, aac, flac, m2a, sdlxiff, ttx, ttml
- Pli ol 500 mil dungeblaj tradukistoj kaj 280 lingvoj;
- API por aliro al eksteraj programoj;
- Ebligas tradukadon de teksto, video, retejo, programo, kaj kursoj de e-lernado;
- Optika rekono de signoj en bildoj kaj .pdf dosieroj;
- Kunlaboro de pluraj tradukistoj samtempe.

Malavantaĝoj:

- Pro la maniero kiel la retejo estas organizita, estas iom malfacile trovi la projektojn, memorojn de tradukado kaj terminarojn;
- La formato de la memoro de tradukado kaj terminaro estas tre simpla, mankas ebleco aldoni metadatenojn;
- Kontrolo de la kvalito de la tradukado estas tre simpla.

6.3 Konkludoj pri la traduk kostoj

Kvankam estas malfacile uzi la NMT-modeloj loke per programoj, la uzo per aliaj platformoj estas facila kaj provizas rapidajn rezultojn. La tradiciaj agentejoj de tradukado daŭre ekzistas, sed nun ili konkuras kun aliaj servoj kiuj provizas simplan, sed rapidajn tradukadojn.

Ni ankoraŭ studas kiel malaltigi la tradukokostojn; kelkaj el la eblecoj inkluzivas:

- Uzi fontodokumentojn kaj agordajn dosierojn bazitaj en la formatoj XML, JSON, YAML, jam kutime uzataj en la entrepreno;
- Kontroli la versiojn de la dokumentoj, memoro de tradukado kaj terminaro per GIT;
- Kiam parto de la dokumento ŝanĝiĝas, sendi, al la maŝina tradukado kaj tradukistoj, nur la segmentojn, kiuj bezonas esti tradukitaj;
- Provi uzi kontrolitan lingvon kaj limigitan nombron da kodopunktoj por faciligi la ĵetonigadon en la NMT-modeloj;

- Plibonigi la uzon de la terminaro por ke ĝi enhavu ne nur teknikajn terminojn kaj akronimojn de la dokumentita programo, sed ankaŭ la elementojn kiuj aperas sur la ekrano de tiu programo, ĉar ili devas havi ununuran tradukon tra la dokumento.

Bibliografio

Banerjee, Satanjeev; Lavie, Alon 2005: METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. En *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, paĝoj 65–72, An-Arbaro, Miĉigano, Junio. Association for Computational Linguistics.

Bawden, Rachel; Poinhos, Jonathan; Kogkitsidou, Eleni; Gambette, Philippe; Sagot, Benoît; Gabay, Simon 2022: Automatic Normalisation of Early Modern French. En *Proceedings of the 13th International Conference on Language Resources and Evaluation (LREC)*, paĝoj 3354-3366, Marsejlo, Francio, Junio 20-25.

Berners-Lee, T. 2005: IETF RFC3986 *Uniform Resource Identifier (URI): Generic Syntax*.
<https://datatracker.ietf.org/doc/rfc3986>

Brown, Tom B.; Mann, Benjamin; Ryder, Nick; Subbiah, Melanie; Kaplan, Jared; Dhariwal, Prafulla; Neelakantan, Arvind; Shyam, Pranav; Sastry, Girish; Askell, Amanda; Agarwal, Sandhini; Herbert-Voss, Ariel; Krueger, Gretchen; Henighan, Tom; Child, Rewon; Ramesh, Aditya; M. Ziegler, Daniel; Wu, Jeffrey; Winter, Clemens; Hesse, Christopher; Chen, Mark; Sigler, Eric; Litwin, Mateusz; Gray, Scott; Chess, Benjamin; Clark, Jack; Berner, Christopher; McCandlish, Sam; Radford, Alec; Sutskever, Ilya; Amodei, Dario 2020: *Language Models are Few-Shot Learners*.

<https://doi.org/10.48550/arXiv.2005.14165>

Bulté, Bram; Vandeghinste, Vincent; Sevens, Leen; Schuurman, Ineke; Van Eynde, Frank 2021: Can Pictograph Translation Technologies Facilitate Communication and Integration in Migration Settings?. *Computational Linguistics in the Netherlands Journal*, 11, paĝoj 189–212.

<https://www.clinjournal.org/clinj/article/view/136>

Cornelius, Eleanor 2016: Potential impact of QT21. En *Proceedings of the 38th Conference Translating and the Computer*, paĝoj 10-18, Londono, Britio, Novembro 17-18. AsLing.

Cybenko, Georg 1989: Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 1989, 2 (4), pp.303-314. 10.1007/BF02551274. hal-03753170

<https://hal.science/hal-03753170/file/Cybenko1989.pdf>

Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina 2019: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. En *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, paĝoj 4171–4186, Mineapolo, Minesoto. Association for Computational Linguistics.

<https://doi.org/10.48550/arXiv.1810.04805>

DITA v 1.3 2018:

<http://docs.oasis-open.org/dita/dita/v1.3/dita-v1.3-part3-all-inclusive.html>

Doddington, George 2002: Automatic Evaluation of Machine Translation Quality using N-gram Co-occurrence Statistics. En *Proceedings of 2nd Human Language Technologies Conference (HLT-02)*, paĝoj 128-132. San-Diego, Kalifornio.

ECMA-404 *The JSON data interchange syntax* 2nd edition, Decembro 2017:

<https://www.json.org/json-eo.html>

Eisenstein, Jacob 2013: What to do about bad language on the internet, *Proceedings of NAACLHLT 2013, Association for Computational Linguistics*, Atlanto, Georgio, pp. 359–369.

EMT *European Master's in Translation - Competence Framework* 2022:

https://commission.europa.eu/system/files/2022-11/emt_competence_fwk_2022_en.pdf

Filip, David 2016: Why XLIFF and Why XLIFF 2? En *Proceedings of the 38th Conference Translating and the Computer*, paĝoj 53-68, Londono, Britio, Novembro 17-18. AsLing.

Gage, Philip 1994: A New Algorithm for Data Compression. *The C Users Journal*. 12(2):23-38, Februaro.

Gambier, Yves 2009: *Competences for professional translators, experts in multilingual and multimedia communication*. EMT expert group, Bruselo, Januaro 2009.

<https://www.scribd.com/document/356704637/emt-competences-translators-en-pdf>

Gledhill, Christopher; Zimina, Maria 2019: The Impact of Machine Translation on a Masters Course in Web Translation: From Disrupted Practice to a Qualitative Translation/Revision Workflow. En *Proceedings of the 41st Conference Translating and the Computer*, paĝoj 60-73, Londono, Britio, Novembro 21-22. AsLing.

Gripenberg, Gustaf 2003: Approximation by neural networks with a bounded number of nodes at each level. *Journal of Approximation Theory*, vol. 122, num. 2, Junio 2003, paĝoj 260-266.

[https://doi:10.1016/S0021-9045\(03\)00078-9](https://doi:10.1016/S0021-9045(03)00078-9)

Hornik, Kurt; Stinchcombe, Maxwell; White, Halbert 1989: Multilayer feedforward networks are universal approximators. *Neural Networks* vol. 2, num. 5, 1989, paĝoj 359-366.

[https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)

Hutchins, W. John 2004: The Georgetown-IBM experiment demonstrated in January 1954. *Machine translation: from real users to research: 6th conference of the Association for Machine Translation in the Americas*, AMTA 2004, Vaŝingtono, DK, Septembro 28 – Oktobro 2, 2004; ed. Robert E. Frederking kaj Kathryn B. Taylor.

ISO Standard 30042 2019: *Management of terminology resources – TermBase eXchange (TBX)*. International Organization for Standardization.

Kudo, Taku 2018a: Subword regularization: Improving neural network translation models with multiple subword candidates. En *Proceedings of the ACL*.

<https://aclanthology.org/P18-1007.pdf>

Kudo, Taku; Richardson, John 2018b: SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. En *Proceedings of the ACL*.

<https://aclanthology.org/D18-2012.pdf>

Lan, Zhenzhong; Chen, Mingda; Goodman, Sebastian; Gimpel, Kevin; Sharma, Piyush; Soricut, Radu 2020: ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *8th International Conference on Learning Representations (ICLR 2020)*. Adis Abebo, Etiopio, Aprilo 26-30, 2020.

<https://doi.org/10.48550/arXiv.1909.11942>

Levenshtein, Vladimir 1966: Binary Codes Capable of Correcting Deletions and Insertions and Reversals. *Soviet Physics - Doklady*, vol. 10, num. 8, paĝoj 707–710.

Lommel, Arle 2016: Blues for BLEU: Reconsidering the Validity of Reference-Based MT Evaluation. En *Proceedings of the LREC 2016 Workshop “Translation Evaluation – From Fragmented Tools and Data Sets to an Integrated Ecosystem”*, paĝoj 63-70, Portorož, Slovenio, Majo 24.

Papineni, Kishore; Roukos, Salim; Ward, Todd; Zhu, Wei-Jing 2002: BLEU: a Method for Automatic Evaluation of Machine Translation. En *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, paĝoj 311–318, Filadelfio, Pensilvanio, USONO, Julio. Association for Computational Linguistics.

Pastor, Gloria Corpas 2020: Language Technology for Interpreters: The Vip Project . En *Proceedings of the 42nd Conference Translating and the Computer*, paĝoj 36-48, per TTT, novembro 18-20. AsLing.

Raffel, Colin; Shazeer, Noam; Roberts, Adam; Lee, Katherine; Narang, Sharan; Matena, Michael; Zhou, Yanqi; Li, Wei; Liu, Peter J. 2020: *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. *Journal of Machine Learning Research*, 21(140):1–67.

<https://doi.org/10.48550/arXiv.1910.10683>

Schubert, Klaus 1986: Linguistic and Extra-Linguistic Knowledge. En *Computers and Translation* 1: 125-152.

Schuster, Mike; Nakajima, Kaisuke 2012: Japanese and Korean Voice Search. 2012 IEEE *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5149-5152.

Sennrich, Rico; Haddow, Barry; Birch, Alexandra 2016a: Neural Machine Translation of Rare Words with Subword Units. En *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (ACL 2016), Berlino, Germanio.

<https://doi.org/10.48550/arXiv.1508.07909>

Sennrich, Rico; Haddow, Barry; Birch, Alexandra 2016b: *Improving Neural Machine Translation Models with Monolingual Data*.

<https://doi.org/10.48550/arXiv.1511.06709>

Sun, Yu; Wang, Shuohuan; Li, Yukun; Feng, Shikun; Chen, Xuyi; Zhang, Han; Tian, Xin; Zhu, Danxiang; Tian, Hao; Wu, Hua 2019: *ERNIE: Enhanced Representation through Knowledge Integration*.

<https://doi.org/10.48550/arXiv.1904.09223>

Tjong Kim Sang, Erik; Bollmann, Marcel; Boschker, Remko ; Casacuberta, Francisco; Dietz, Feike; Dipper, Stefanie; Domingo, Miguel; van der Goot, Rob; van Koppen, Marjo; Ljubesic, Nikola; Ostling, Robert; Petran, Florian; Pettersson, Eva; Scherrer, Yves; Schraagen, Marijn; Sevens, Leen; Tiedemann, Jorg; Vanallemeersch, Tom; Zervanou, Kalliopi 2017: The CLIN27 Shared Task: Translating Historical Text to Contemporary Language for Improving Automatic Linguistic Annotation. *CLIN Computational Linguistics in the Netherlands Journal*, 7, paĝoj 53-64.

<https://www.clinjournal.org/clinj/article/view/68>

TMX 1.4b *Translation Memory eXchange*

<https://www.gala-global.org/tmx-14b>

Tytler, Alexander Fraser (Lord Woodhouselee) 1813: *Essay on the Principles of Translation*.

Tria eldono. Gutenberg Project, 2021-03-20 [eBook #64890]

<https://www.gutenberg.org/ebooks/64890>

Unicode 14.0 *Character Code Charts*

<https://www.unicode.org/charts>

Weaver, Warren 1949: Translation. En *Machine translation of languages: fourteen essays*, paĝoj 15-23. William N. Locke kaj A. Donald Booth, Technology Press of the Massachusetts Institute of Technology, Cambridge, Mass., kaj John Wiley & Sons, Inc., Novjorko, 1955.

Witkam, A. P. M. 1983: *DLT Distributed Language Translation – a multilingual system for computer networks*. BSO Buro voor Systeemontwikkeling, Utreĥto.

XCOPA A *Multilingual Dataset for Causal Commonsense Reasoning*

<https://github.com/cambridgeltl/xcopa>

XLIFF 1.2 *XML Localization Interchange File Format Version 1.2*. 2008.

<https://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html>

XLIFF-2.1 *XML Localization Interchange File Format Version 2.1*. Eldono de David Filip, Tom Comerford, Soroush Saadatfar, Felix Sasaki, kaj Yves Savourel. 2018-02-13:
<http://docs.oasis-open.org/xliff/xliff-core/v2.1/os/xliff-core-v2.1-os.html>

XML Extensible Markup Language (XML) 1.0 (Fifth Edition)
<https://www.w3.org/TR/REC-xml/>

Yang, Zhilin; Dai, Zihang; Yang, Yiming; Carbonell, Jaime; Salakhutdinov, Ruslan; Le, Quoc V. 2020: *XLNet: Generalized Autoregressive Pretraining for Language Understanding*.
<https://doi.org/10.48550/arXiv.1906.08237>

Yergeau, F. 2003: IETF RFC3629 *UTF-8, a transformation format of ISO 10646*.
<https://datatracker.ietf.org/doc/rfc3629>

Resumo

Aŭtomataj Tradukiloj

La celo de ĉi tiu esplorado estis studi la ekzistantajn aŭtomatajn tradukilojn kaj serĉi eblecojn por malaltigi la kostojn de tradukado de teknikaj dokumentoj, kiu hodiaŭ estas farita de ekstera tradukagentejo.

Unue ni esploris la ŝanĝojn en la postuloj, kiuj okazis en la fako de tradukado. Poste, DITA estis elektita kiel la plej taŭga formato de la fontodokumentoj, kiuj estas uzataj por generi la celdokumentojn en pluraj elekteblaj formatoj, ebligante ankaŭ administri kaj kontroli la kvaliton de la tuta traktado per GIT. Fine ni studis kiajn tradukmodelojn ekzistas kaj kiel ili funkcias.

Inter la eltrovoj ni malkovris ke la laboro de tradukistoj ŝanĝiĝas pro la avantaĝo de tuja unua traduko provizita de la maŝina tradukado, sed samtempe malpliigas simplan tradukadon pro la konkurenco kun la maŝina tradukado mem, kiam la postulo estas nur rapida kaj esenca traduko. Aliflanke, novaj laborŝancoj aperas por tradukistoj pro la bezono agordi kaj prepari datenojn uzatajn en la evoluo de MT-modeloj.

Artefarita intelekto ŝanĝis kiel homoj uzas komputilojn por solvi problemojn, pro tio ankaŭ la metio de tradukado ŝanĝiĝis. Tradukistoj devas havi altnivelan scipovon kaj adapteblecon al teknologiaj ŝanĝoj. Uzo de kvalifitcertigiloj kaj administradiloj de la traduklaborfluo nun estas parto de la procezo de tradukado.

Aliro al la interreto ebligas fari ajnan intelektan laboron ie ajn en la mondo, se la necesaj konoj estas haveblaj, ĉar la teknikaj rimedoj kutime estas interrete alireblaj, pro tio la konkurenco inter entreprenoj kaj inter profesiuloj pligrandiĝis.

La ekesto de retaj platformoj por tradukado estas uno el la kaŭzoj de ĉi tiu konkurenco, pro kiu ili provizas ŝancon por malaltigi la tradukokostojn. Sed, por ke la provizitaj servoj estu same kvalitaj kiel tiuj provizitaj por tradiciaj tradukagentejoj necesas efektivigi kvalitan certigan laborfluan kaj uzi administradilojn.

Tial, ni antaŭvidas eblecon por la uzo de hibrida modelo, kie kontrolo de la dokumentoj, tradukmemoro kaj terminaro estas farita per konataj pli precizaj reguloj, kaj la tradukado farita de NMT-modelo estas kontrolita de tradukistoj.

Eĉ se la interna uzo de maŝina tradukado ne estis elektita, la studo de la apliko de artefarita intelekto en aliaj kampoj estas ebleco por la daŭrigo de esplorado kaj evoluo de la spertoj akiritaj en ĉi tiu studo kaj certe produktos bonajn rezultojn.

Summary

Automatic Translation Tools

This research has been conducted with the aim of studying existing machine translation tools and exploring possibilities to reduce the costs of translating technical documents, which are currently handled by an external translation agency.

At first, we examined the changes in requirements that have occurred in the translation field. Then, DITA was selected as the most suitable format for the source documents, which are used to generate target documents in various selectable formats, also allowing the management and quality control of the entire process via the GIT platform. Finally, we studied the existing translation models and how they work.

Among the findings, we discovered that the role of translators is changing due to the advantage of instant initial translation provided by machine translation. However, it also decreases the demand for simple translations due to competition with machine translation itself, when only a quick and gist translation is needed. On the other hand, new job opportunities are emerging for translators due to the need to configure and prepare data used in the development of machine translation models.

Artificial intelligence has changed how people use computers to solve problems, which has also transformed the translation profession. Translators must possess advanced skills and adaptability to technological changes. Additionally, the use of quality assurance tools and workflow management tools is now part of the translation process.

Internet access allows any intellectual work to be done anywhere in the world, provided the necessary knowledge is available, as the technical resources are typically accessible online. This possibility has increased competition among companies and professionals.

The emergence of online translation platforms is one of the causes of this competition, offering an opportunity to reduce translation costs. However, to ensure that the services provided are of the same quality as those offered by traditional translation agencies, it is necessary to implement a quality assurance workflow and use management tools.

Therefore, we foresee the possibility of using a hybrid model, where the control of documents, translation memory, and terminology is handled by more precise, established rules, and the translation performed by a neural machine translation model is reviewed by translators.

Even though the internal use of machine translation has not been chosen, studying the application of artificial intelligence in other fields presents an opportunity for continued research and development of the experience obtained in this study, and will certainly yield tangible results.

Sumário

Ferramentas de Tradução Automática

O objetivo desta pesquisa foi estudar os programas existentes para tradução automática e explorar possibilidades para reduzir os custos de tradução de documentos técnicos, que atualmente são feitos por uma agência de tradução externa.

Primeiro, exploramos as mudanças nas exigências que ocorreram no campo da tradução. Em seguida, DITA foi escolhido como o formato mais adequado para os documentos fonte, que são usados para gerar os documentos finais em vários formatos elegíveis, permitindo também gerenciar e controlar a qualidade de todo o processo por meio da plataforma GIT. Finalmente, estudamos quais modelos de tradução existem e como eles funcionam.

Entre as descobertas, constatamos que o trabalho dos tradutores está mudando devido à vantagem de uma primeira tradução instantânea fornecida pelas ferramentas de tradução automática, mas que reduz, ao mesmo tempo, a demanda por traduções simples devido à concorrência com a própria tradução automática, quando a necessidade é apenas uma tradução rápida para uma visão geral. Por outro lado, novas oportunidades de trabalho surgem para tradutores devido à necessidade de configurar e preparar dados usados no desenvolvimento de modelos de tradução automática.

A inteligência artificial mudou a forma como as pessoas usam computadores para resolver problemas, e isso também transformou a profissão de tradução. Os tradutores precisam ter habilidades avançadas e adaptabilidade às mudanças tecnológicas. Adicionalmente, o uso de ferramentas de garantia da qualidade e de gestão do fluxo de trabalho agora faz parte do processo de tradução.

Acesso à internet permite que qualquer trabalho intelectual seja realizado a partir de qualquer lugar do mundo, desde que o conhecimento necessário esteja disponível, pois os recursos técnicos geralmente são acessíveis remotamente. Por isso, a concorrência entre empresas e profissionais aumentou.

O surgimento de plataformas remotas de tradução é uma das causas dessa concorrência, oferecendo oportunidade para reduzir os custos de tradução. No entanto, para garantir que os serviços prestados sejam da mesma qualidade que os oferecidos pelas agências de tradução tradicionais, é necessário implementar um fluxo de trabalho de garantia da qualidade e usar ferramentas de gestão.

Portanto, prevemos a possibilidade de usar um modelo híbrido, onde o controle dos documentos, da memória de tradução e da terminologia são geridos por um sistema de regras precisas e conhecidas, e a tradução, feita por um modelo de tradução automática, é revisada por tradutores.

Mesmo que o uso interno de tradução automática não tenha sido escolhido, o estudo da aplicação da inteligência artificial em outros campos apresenta uma oportunidade para a continuidade da pesquisa e do desenvolvimento das experiências adquiridas neste estudo e certamente produzirá resultados proveitosos.