Институт за математику и информатику
Природно-математички факултет
Универзитет у Крагујевцу

Мастер рад

Примена рекурентних неуронских мрежа у обради природних језика

Студент:
Драгутин Остојић

Професор:
др Татјана Стојановић

Август 2020.

Захваљујем се свом ментору др Татјани Стојановић и колеги Бранку Арсићу на великој помоћи у току реализације овог мастер рада, као и др Милошу Ивановићу на обезбеђивању неопходних хардверских ресурса.
Посебну захвалност дугујем Анђелки Пантовић за помоћ на лектури и преводу.

Садржај

Глава 1

Увод

Природни језици подразумевају језике који су спонтано настали у давним временима и којима говоре поједине људске заједнице (народи, нације, племена). Поред природних језика, постоје и плански језици који се користе или се могу користити за споразумевање, као што су есперанто, интерлингва и идо. Такође, постоје и вештачки језици створени из научних разлога као језички експеримент или из практичних разлога као што су програмски језици, фиктивни и тајни језици [55].

Још од средњег века датирају тежње људи за формализацијом природних језика. У деветнаестом веку обнавља се интересовање за ову област, док у двадесетом веку нагли развој математичке логике поспешује нова истраживања у домену опште лингвистике. Велику прекретницу представља књига Чомског *(енгл. Noam Chomsky)* из 1957. године под називом *Syntactic structures* [15]. Иако тада проучаване формалне граматике нису успеле да до краја опишу говорни језик људи, њихово изучавање дало је велики допринос у развоју програмских језика.

У двадесетом веку јављају се и прва достигнућа у аутоматској обради природних језика захваљујући наглом развоју рачунара. У почетку, обрада се фокусирала углавном на текст, а неки од задатака које је требало аутоматизовати су класификација, сентимент анализа, препознавање језика и дијалекта, аутоматско комплетирање и многи други. Међу бројним проблемима ове врсте налази се и задатак аутоматског превођења са једног на други језик. Због велике потребе за таквим достигнућем и практичне применљивости која би олакшала живот у многим аспектима, овај проблем се интензивно проучава до данас, а област носи назив машинско превођење.

И поред бројних достигнућа и различитих техника које су развијене у дугом временском периоду, преводи машинских преводилаца нису били упоредиви по квалитету са преводима људи, иако се још педесетих година тврдило да смо надомак тога.

Неуронске мреже, као модел надгледаног машинског учења, због своје специфичности да саме екстрактују особине из датих података, показале су се као врло ефикасне при решавању проблема код којих је јако тешко издвојити скуп правила која проблем потпуно описују. Један од таквих проблема је и превођење између два језика.

Рекурентне неуронске мреже су због својих способности да обрађују низове података биле најприродније решење за проблем превођења. Међутим, услед потешкоћа са њиховим тренирањем, деценијама није било значајнијих помака. Тек пре пар година појавиле су се идеје, а и хардверске могућности за њихову реализацију, да се машинско превођење у пракси потпуно пребаци на терен дубоког машинског учења. Ова техника показала се као изузетно успешна, а као врло млада поседује велики потенцијал да временом постигне још боље резултате.

Машинско превођење изучавано је највише на језицима за које је постојала највећа практична потреба да се међусобно преводе. Све је почело са руским и енглеским језиком, касније су се укључили француски, јапански, а онда и кинески и остали. Међутим, ова

област јако мало је примењивана на јужнословенским језицима. Разлози за то могу се тражити у јако скупом развоју неких старијих система и у недостатку великих корпуса, неопходних за новије системе машинских преводилаца. Нови приступ који подразумева употребу неуронских мрежа, знатно отклања ове препреке. Сам развој није скуп, чему доприноси и отвореност софтверских технологија које се користе, док количина података потребних за добре резулате не мора бити велика као некад.

У овом раду описан је поступак израде неуронског машинског преводиоца, који преводи са енглеског на српски језик. Приликом израде имплементирана су последња достигнућа у овој области, с тим што је потенцирано задржавање једноставности модела и могућности да се он обучи у разумном времену чак и на кућном рачунару. Акценат је стављен на чисту и прошириву имплементацију, а избегаване су библиотеке високог нивоа апстракције. Детаљно су описани разни изазови у креирању модела, а посебно осетљиво питање одабира, прикупљања и пречишћавања података.

Рад подразумева елементарно познавање принципа функционисања неуронских мрежа као и језика *Bash* и *Python*. У глави 2 описан је принцип рада рекурентних неуронских мрежа у више различитих варијанти које се користе. Глава 3 бави се разним приступима машинском превођењу кроз историју. Глава 4 описује аутентичан начин припреме корпуса за овај задатак, што представља веома значајан фактор у крајњем резултату. У последњој глави 5, представљени су резултати спроведених експеримената и њихово поређење са резултатима које постижу решења познатих корпорација.

Циљ овог рада је сагледавање могућности и захтева тренутно доступне технологије и постављање основе за даља истраживања у области неуронског машинског превођења за енглеско-српски превод.

Глава 2

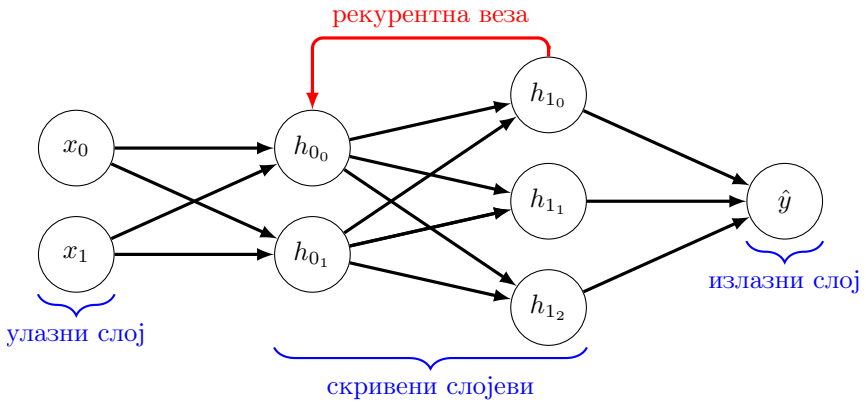
Рекурентне неуронске мреже

Рекурентна неуронска мрежа *(енгл. Recurrent Neural Network – RNN)* је свака вештачка неуронска мрежа која садржи циклус [34]. Ова особина им даје могућност да показују временски динамично понашање. За разлику од вишеслојних перцептрона *(енгл. Multilayer Perceptron – MLP)*, који се још називају и мреже са пропагацијом унапред *(енгл. Feedforward Neural Networks)*, рекурентне неуронске мреже могу користити своје унутрашње стање (меморију) за обраду секвенци улаза. Ово их чини применљивим на задатке као што су препознавање рукописа, препознавање говора, компоновање музике и још много тога.

Термин рекурентна неуронска мрежа упућује на две широке класе неуронских мрежа са сличном општом структуром, где је једна коначни импулс, а друга бесконачни импулс. Обе класе показују временски динамично понашање [57]. Коначна импулсна RNN је усмерени ациклични граф који се може размотати *(енгл. unfolding)* и заменити MLP-ом, а бесконачна импулсна RNN је усмерени циклични граф који се не може одмотати.

Обе врсте RNN-а могу имати додатно складишно стање, а складиштење може бити под директном контролом неуронске мреже. Таква контролисана стања називају се капије, и она су саставни елементи дуге краткорочне меморије *(енгл. Long Short-Term Memory – LSTM)*. На слици 2.1 приказана је једноставна архитектура RNN са улазним и излазним слојем неурона, где црвена стрелица означава рекурентну везу која излаз вишег слоја враћа као улаз нижем слоју.

Рекурентне неуронске мреже увео је Румелхарт *(енгл. David E. Rumelhart)* 1986. године [75], а LSTM Хочрајтер *(нем. Sepp Hochreiter)* и Шмитхубер *(нем. Jürgen Schmidhuber)* 1997. године [28]. Данас се у различитим областима проучава и користи широк спектар архитектура RNN-а.



Слика 2.1: Пример RNN (црвена веза мрежу чини рекурентном)

2.1 Концепт RNN-а

2.1.1 Секвенце података

Механизам по коме раде RNN јако је погодан за моделовање секвенци података. У циљу бољег разумевања, биће представљено неколико кратких експеримената [66]. Нека је на слици 2.2 дата лопта која се креће.

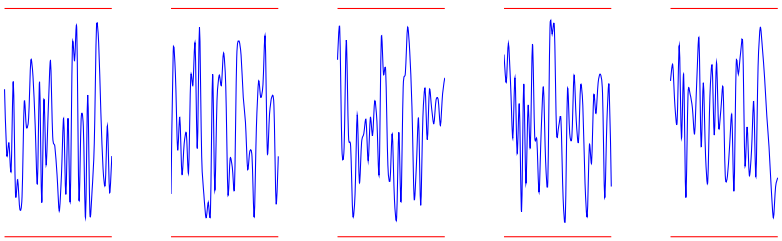
Слика 2.2: Лопта која се креће

Потребно је одредити правац кретање лопте. Сам податак који се може добити са слике 2.2, није довољан при доношењу закључка о њеном правцу кретања. Без знања о томе где је лопта била, нема довољно информација да би се предвидело куда она иде.

Међутим, ако постоји секвенца података о више узастопних позиција лопте, као на слици 2.3, онда је већ могуће направити прогнозу са одређеном прецизношћу и донети закључак да се она помера на десно.

Слика 2.3: Лопта која се креће са претходним позицијама

Секвенцијални подаци појављују се у многим облицима. Аудио спектограм могуће је поделити на мале временске интервале, као на слици 2.4, и они се могу посматрати као секвенца података.



Слика 2.4: Аудио спектограм подељен на узастопне делове

Текст се може поделити на секвенце карактера, слогова, речи, реченица, фраза... Без знања јасних правила за границе секвенци, текст је тешко тумачити. Пример поделе на речи реченице написане на јапанском језику дат на слици 2.5.

住宅地域|に|おける| 本機|の |使
用 |は|有害な|電波妨害|を|引き起
こす|こと|が|あり、|その|場合|ユー
ザー|は|自己負担|で|電波妨害|の|問
題|を|解決|しなければなりません。

Слика 2.5: Реченица написана сложеним јапанским писмом подељена на речи

2.1.2 Секвенцијална меморија

RNN се користе за предикцију на основу обраде секвенци, при чему кључну улогу игра секвенцијална меморија. Да би се интуитивно стекао увид у појам секвенцијалне меморије, биће направљен осврт на начин на који људи размишљају.

Дат је следећи експеримент: Испитаник (човек) треба да понови азбуку, дату на слици 2.6, без гледања у њен запис.

А Б В Г Д Ђ Е Ж З И Ј К Л Љ М Н Њ О П Р С Т Ћ У Ф Х Ц Ч Џ Ш

Слика 2.6: Азбука

То би требало да буде веома лако. Ако је испитаник ову секвенцу већ знао, било би му потребно јако мало времена да је понови.

Колико би му било тешко да на пример крене од слова Ф? На почетку имао би проблема са првих пар слова, али након тога мозак ће препознати образац и остатак ће ићи течно.

У наредном експерименту испитаник треба опет да понови азбуку, али сада у обрнутом поретку, као на слици 2.7.

Ш Џ Ч Ц Х Ф У Ћ Т С Р П О Њ Н М Љ Л К Ј И З Ж Е Ђ Д Г В Б А

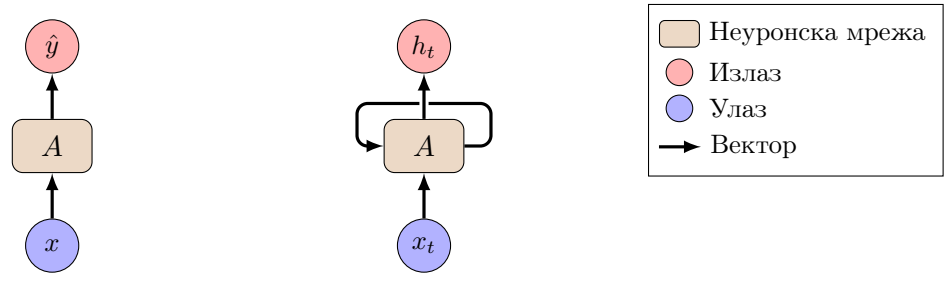
Слика 2.7: Азбука у обрнутом редоследу

То је већ знатно тежи задатак. Осим ако то није радио раније, за понављање ове секвенце било би му потребно знатно више времена.

Постоји веома логичан разлог зашто је први задатак много лакши од друга два. Људи уче алфабет као секвенцу. Секвенцијална меморија је механизам који олакшава мозгу да препознаје обрасце. На сличном принципу раде и рекурентне неуронске мреже.

Људи не покрећу нови мисаони процес у свакој новонасталој ситуацији. Читањем овог текста разумевање сваке речи произилази на основу разумевања претходних речи. Када нова реч бива прочитана не одбацују се све претходне и не почињете процес размишљања испочетка. Мисли имају доследност. Традиционалне неуронске мреже не поседују сличну способност и чини се да им је то главни недостатак.

На пример, нека постоји потреба да се класификују врсте догађаја које се дешавају у сваком тренутку једног филма. Нејасно је како би традиционална неуронска мрежа могла да користи своје резоновање о претходним догађајима у филму како би на основу њих закључила смисао будућих догађаја. RNN се баве овим питањем. Оне садрже циклусе у себи, што им омогућава задржавање информација.



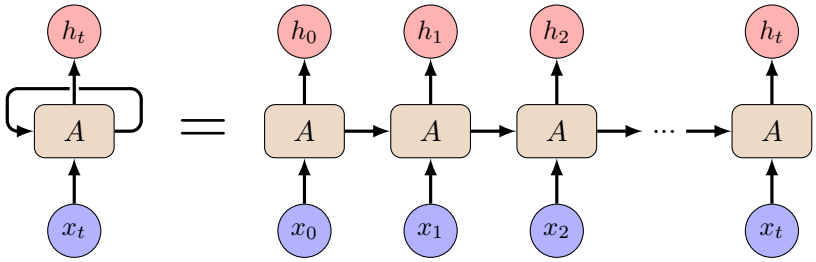
(а) MLP (б) RNN

Слика 2.8: Разлика између MLP и RNN

На слици 2.8а на једноставан начин представљена је структура традиционалног MLP-а, који прима улазни вектор података x и на основу неуронске мреже A врши предвиђање

излазног вектора $\hat{y}$. На сличан начин на слици 2.8б неуронска мрежа A узима улаз x_t и генерише излаз h_t, са тим што јој циклус омогућава да се добијене информације пренесу и у следећи временски корак, где се под временским кораком подразумева време потребно за формирање излаза из мреже на основу једног улаза.

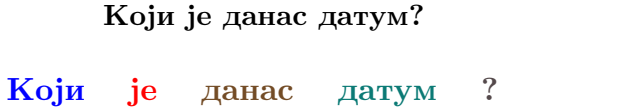
Циклуси RNN чине тежом за посматрање и анализу, међутим, она се може приказати и као низ копија једне исте мреже, где свака претходна преноси поруку следећој.



Слика 2.9: Размотани приказ неуронске мреже

Нека је дат пример чет робота, софтвера који примењује RNN и који се данас често користи. Робот треба, на основу текста који је примио од корисника, да закључи шта корисник жели. Да би се то постигло, прво треба кодирати секвенцу текста користећи RNN, а затим тај излаз убацити у MLP који ће класификовати изражену намеру. На пример, од корисника се добија реченица – *Који је данас датум?* .

Пре свега треба издвојити речи из реченице и једну по једну проследити рекурентној неуронској мрежи, као на слици 2.10.

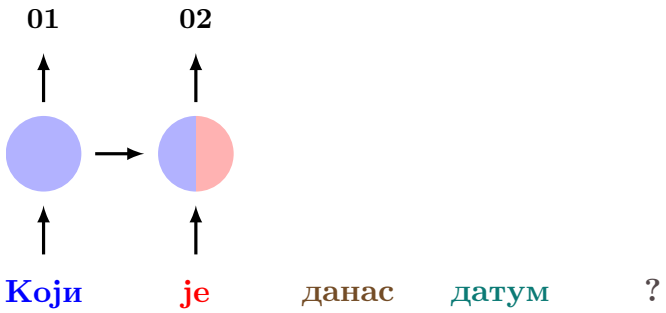


Слика 2.10: Подела реченице на речи

Наредни корак је слање речи *Koju* на улаз RNN, која ће после једне итерације дати први излаз, као на слици 2.11.

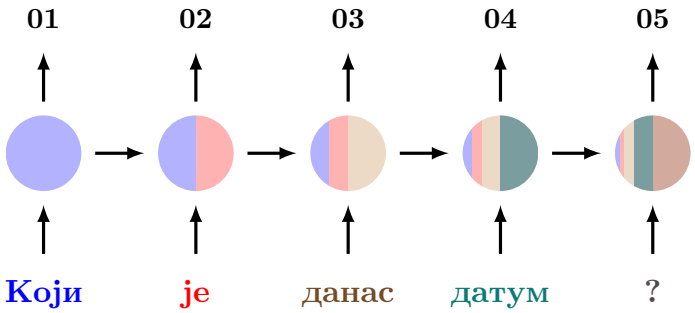
Слика 2.11: Поступак формирања коначног излаза RNN-а

У следећем кораку на улаз доводи се реч *је*, али се прослеђује и излаз претходног корака, као на слици 2.12. RNN сада има информације о речима *Koju* и *је*.



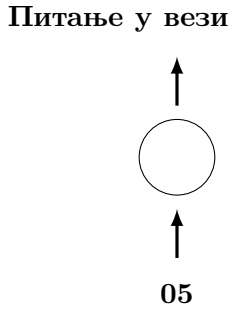
Слика 2.12: Слање друге речи и првог излаза на улаз RNN-а

Процес се понавља све до последње речи. У завршном кораку, на слици 2.13, може се уочити стање RNN-а које има кодиране информације о свим речима из претходних корака.



Слика 2.13: Поступак формирања коначног излаза RNN-а

Пошто је коначни излаз креиран на основу читаве реченице, он се у наредном кораку шаље на улаз MLP-а, који је задужен да на основу њега класификује намеру корисника, што је приказано на слици 2.14.



Слика 2.14: Слање излаза MLP мрежи на класификацију

По представљеном принципу може се решавати широка класа различитих проблема. RNN у оваквом систему има задатак акумулације знања садржаног у секвенци података, које уједно и енкодира, а потом наредни систем врши даље процесирање на основу тако добијених информација.

Примери су: препознавање говора, класификација осећања, анализа ДНК секвенци, машинско превођење, препознавање активности на видео снимку, препознавање ентитета у тексту, генерисање описа фотографија...

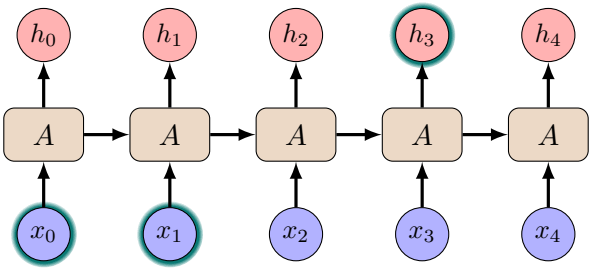
2.1.3 Проблем дугорочних зависности

Добра особина RNN-а је њихова способност да повежу претходне информације са садашњим задатком. Међутим, расподела боја на слици 2.15 илуструје један озбиљан проблем ове архитектуре познат као нестајање градијента *(енгл. Vanishing Gradient)*.

Слика 2.15: Коначно скривено стање RNN-а

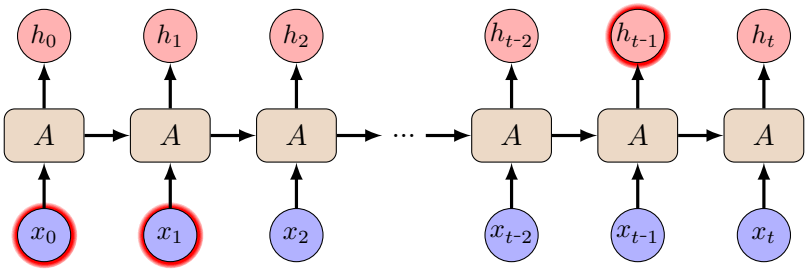
Краткорочна меморија *(енгл. Short-Term Memory)*, као таква, директно је узрокована овим проблемом, који је присутан и у другим врстама неуронских мрежа. Како RNN обрађује секвенце у више корака, тако има и више проблема са задржавањем старих информација. Као што се може видети на слици 2.15, информација добијена из речи *Који* и *је* није у великој мери присутна у последњем временском кораку. Међутим, кратко питање, какво је постављено чет роботу, RNN ипак може разумети. Понекад је довољно поседовати само најсвежије податке да би актуелни задатак био обављен.

Нека је дат језички модел који предвиђа следеће речи на основу претходних [61]. Ако треба предвидети наредну реч у изразу *Облаци су на* _____, није потребан никакав шири контекст, прилично је јасно да ће следећа реч бити *небу*. У оваквим случајевима, где је мала удаљеност између релевантне информације и места на коме је потребно направити закључак на основу ње, RNN може научити да користи ту информацију. Пример овакве ситуације илустрован је на слици 2.16.



Слика 2.16: Релевантна информација у близини тренутног контекста

Међутим, постоје и примери у којима је потребан шири контекст да би се нешто закључило. У примеру *Одрастао сам у Француској, ... Течно говорим* _____ ужи контекст налаже да је следећа реч највероватније име неког језика, али ако треба одредити који језик је у питању, потребан је и контекст *Француска*, из удаљене претходне реченице. У потпуности је могуће да се размак између релевантне информације и места на ком је она тражена знатно прошири, што је илустовано на слици 2.17. Како размак све више расте, тако и RNN има све мању способност да научи да повеже информације.



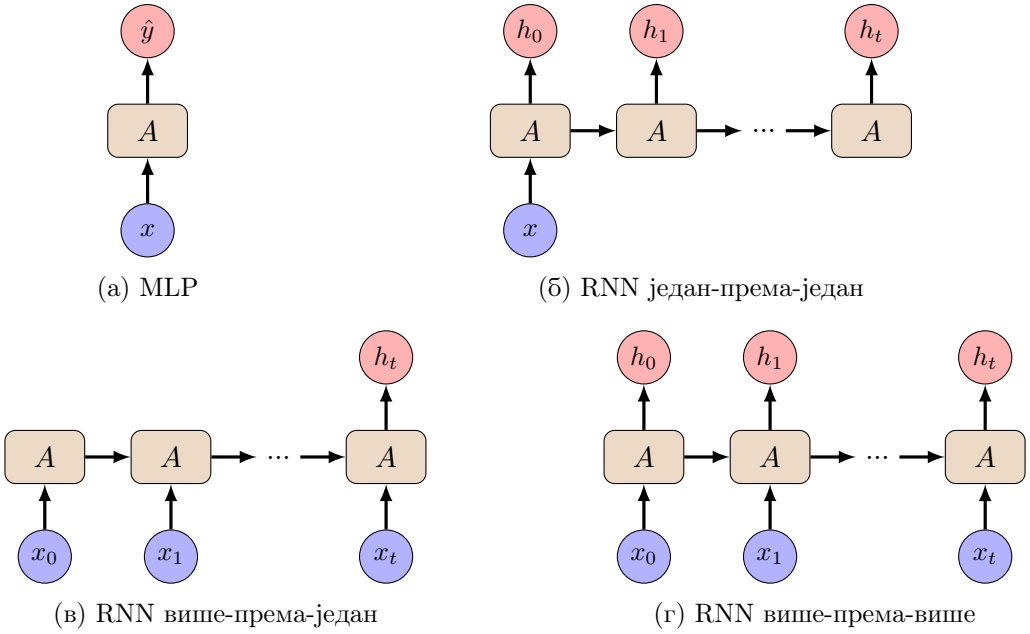
Слика 2.17: Релевантна информација удаљена од тренутног контекста

У теорији човек би могао пажљиво изабрати параметре по којима ће RNN у датим проблемима учити дугорочне зависности. Нажалост, у пракси, класичне RNN не изгледају као да могу да их науче. Проблем су истраживали 1991. године Хочрајтер [27] и 1994. године Бенџио *(франц. Yoshua Bengio)* [7], који су пронашли фундаменталне разлоге због којих би то било веома тешко.

2.2 Модел RNN-а

2.2.1 Архитектура

Један од проблема са MLP-ом (и још неким типовима неуронских мрежа) је да оне раде само са унапред одређеним величинама. Узимају улазе фиксне величине и производе излазе фиксне величине. RNN су другачије јер дозвољавају секвенце улаза и излаза променљивих дужина. На слици 2.18 илустровани су MLP и неколико генералних архитектура RNN.



(а) MLP

(б) RNN један-према-један

(в) RNN више-према-један

(г) RNN више-према-више

Слика 2.18: MLP и различите RNN архитектуре [35]

Слика 2.18а представља традиционални начин рада неуронске мреже, унос је фиксне величине, излаз такође. Једна од примена ове мреже је класификација фотографија. Слика 2.18б представља RNN архитектуру код које је улаз фиксне величине, док је њен излаз секвенцијалан. Пример примене овакве архитектуре је генерисање описа на основу датих фотографија. На слици 2.18в приказана је варијанта RNN-а са секвенцијалним улазом, док је излаз фиксне величине. Њена примена би могла бити анализа осећања, где одређену реченицу треба класификовати као изражавање позитивне или негативне емоције. Слика 2.18г приказује RNN архитектуру која укључује синхронизовани улаз и излаз секвенци. Пример њене примене био би препознавање дешавања на видео запису, где треба означити сваки кадар.

Значајно је приметити да код свих варијанти RNN-а нема унапред датих ограничења за дужине секвенци које се уносе или генеришу. Неуронска мрежа је фиксна и понављајућа (рекурентна) и може се применити онолико пута колико је потребно.

Осим представљених уопштења постоји још много различитих изведених варијанти, о некима ће бити речи касније. У овом раду биће разматран модел коначног импулса, а подразумевана више-према-више архитектура са слике 2.18г, јер представља у одређеном смислу генерални случај.

RNN имају ланчану форму и сачињене су од неуронских мрежа које се понављају и које носе назив модули, ћелије или кораци. Код стандардних RNN-а, ови понављајући модули имају врло једноставну структуру, а то је најчешће један мрежни слој са *tanh* активационом функцијом.

2.2.2 Осврт на MLP модел

Закључак који мрежа изводи често се представља у виду вероватноће за сваку од K класа које је могуће предвидети. Приступ који се често користи да би се од низа вредности добио низ вероватноћа је функција softmax. Она прихвата вектор $\vec{x} = (x_0, x_2, \ldots, x_{K-1})$ произвољних вредности и пресликава га у вектор који садржи расподелу вероватноћа, са вредностима у распону $(0, 1]$, где је њихов укупан збир 1. Нормализована експоненцијална функција softmax : $\mathbb{R}^K \to (0, 1]^K$ дефинисана је релацијом (2.1).

$$\text{softmax}(\vec{x})_i = \frac{e^{\vec{x}_i}}{\sum_{k=0}^{K-1} e^{\vec{x}_k}} \qquad \forall i \in [0, K-1] \tag{2.1}$$

Тренирање једнослојног MLP-а, врши се пропагацијом уназад *(енгл. Backpropagation – BP)* [75] на датом тренинг скупу $T \subseteq \mathbb{R}^N \times \mathbb{R}^M$, величине $n \in \mathbb{N}$, за парове $(x_i, y_i) \in T$, $0 \leq i < n$, $i \in \mathbb{N}$, где x_i представља дати улаз, а y_i очекивани излаз. Овај алгоритам састоји се од две основне фазе: проласка унапред *(енгл. Forward Pass)* и проласка уназад *(енгл. Backward Pass)*.

Уз увођење ознака:

- W - Тежинска матрица;
- b - Померај *(енгл. bias)* улазног вектора;
- $\hat{y}_i$ - Нормализоване вероватноће излаза.

једнослојни MLP може се описати једначином (2.2) и сликом 2.19.

$$\hat{y}_i = \text{softmax}\left(W x_i + b\right) \tag{2.2}$$

Слика 2.19: Једнослојни MLP са softmax активационом функцијом

Фаза проласка унапред почиње израчунавањем губитака *(енгл. Loss)* $l_i \in \mathbb{R}$, тј. грешке у предвиђању излаза $\hat{y}_i$ за дати улаз x_t у односу на очекивани излаз y_i, што је приказано једначином (2.3).

$$l_i = \text{LossFunction}\left(\hat{y}_i, y_i\right) \tag{2.3}$$

LossFunction : $\mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}$ припада скупу различитих функција које анализирају тачност предвиђања. Завршни део фазе проласка унапред је понављање (2.2) и (2.3) за сваки пар из скупа T, где се сумирањем свих l_i добија укупан губитак *(енгл. Total Loss)*, тј. укупна грешка, $L \in \mathbb{R}$, што приказује једначина (2.4).

$$L = \sum_{k=0}^{n-1} l_i \tag{2.4}$$

Један пролаз кроз тренинг скуп назива се епоха *(енгл. epoch)*. После једне епохе, уз познату вредност L може се прећи на пролазак уназад. У овој фази треба ажурирати W тако да укупни губитак буде смањен. То се ефикасно може постићи градијентним спустом [18], а записано је једначином (2.5).

$$W = W - \mu \frac{\partial L}{\partial W} \tag{2.5}$$

У једначини (2.5) хиперпараметар μ одређује брзину ажурирања *(енгл. Learning rate)*.

Ове две фазе изводе се итеративно за одређени број епоха, све док се не постигну одговарајуће перформансе модела.

Једнослојни MLP са softmax активационом функцијом узет је као пример, пре свега зато што је одређен само једном тежинском матрицом W. Тренирање вишеслојне неуронске мреже врши се на идентичан начин, са тим што једначина која њу описује садржи више тежинских матрица, и више нелинеарних функција.

2.2.3 RNN модел

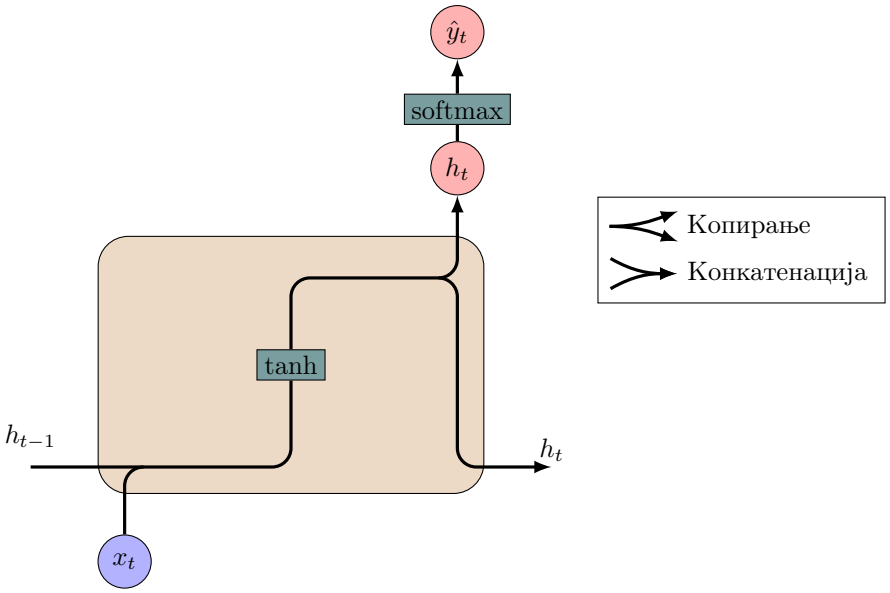
У случају секвенцијалних података, тренинг скуп $T \subseteq \mathcal{P}\left(\mathbb{R}^N\right) \times \mathbb{R}^M$, величине $n \in \mathbb{N}$, садржи парове $(x_i, y_i) \in T$, $0 \leq i < n$, $i \in \mathbb{N}$, где $x_i = (x_{i_0}, x_{i_1}, \ldots, x_{i_t})$ за $t \in \mathbb{N}$ представља једну секвенцу улаза, а y_i очекивани излаз. Тренирање RNN-а, одвија се на сличан начин, алгоритмом који носи назив пропагација у прошлост *(енгл. Backpropagation Through Time – BPTT)* [75].

Уз увођење ознака:

- W_i - Тежинска матрица која параметризује везе улазних вектора;
- W_y - Тежинска матрица која параметризује везе излазног вектора;
- b_i - Померај улазних вектора;
- b_y - Померај излазног вектора;

једноставна RNN архитектура може се описати једначинама (2.6) и сликом 2.20.

$$\begin{aligned} h_t &= \tanh\left(W_i \left[h_{t-1}, x_t\right] + b_i\right) \\ \hat{y}_t &= \operatorname{softmax}\left(W_y h_t + b_y\right) \end{aligned} \tag{2.6}$$



Слика 2.20: Структура RNN ћелије

BPTT ради у две идентичне фазе као и BP. Пролазак унапред почиње прихватањем улаза $x_i = (x_{i_0}, x_{i_1}, \ldots, x_{i_t})$ и чувањем акумулираних информација у скривеном вектору h_t, на основу кога се у било ком моменту може направити предвиђање $\hat{y}_t$. h_0 се иницијализује као нула вектор и ажурира у сваком временском кораку. Процес ажурирања дефинисан је једначинама (2.6). Тотални губитак може се израчунати једначинама (2.3) и (2.4).

Након завршене епохе, потребно је извршити пролазак уназад и притом ажурирати обе тежинске матрице (2.7).

$$\begin{aligned} Wi &= Wi - \mu \frac{\partial L}{\partial Wi} \\ Wy &= Wy - \mu \frac{\partial L}{\partial Wy} \end{aligned} \tag{2.7}$$

Процес треба поновити у одговарајућем броју епоха.

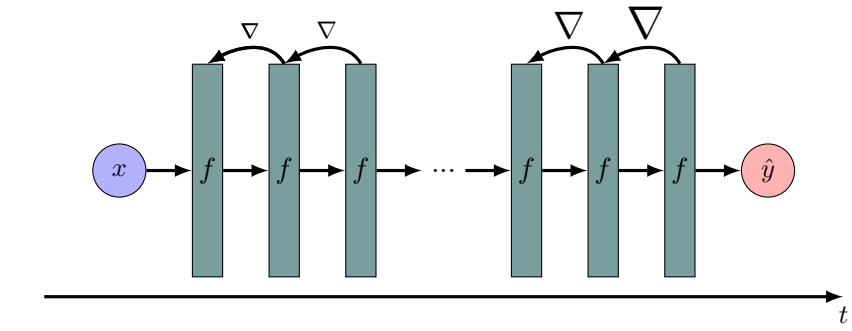
2.2.4 Нестајање и експлозија градијента

Краткорочна меморија и нестајање градијента су уско повезани са природом BP и BPTT алгоритма. Размотавање графа показује да се током једне секвенце рекурентна мрежа може посматрати као MLP са релативно великим бројем надовезаних скривених слојева и релативно великим бројем веза. Различите секвенце које се истовремено налазе у скупу за обучавање даће различите градијенте, али се ти градијенти једноставно сабирају захваљујући томе што је грешка на целом скупу за обучавање просто збир (или просек) грешака на појединачним секвенцама.

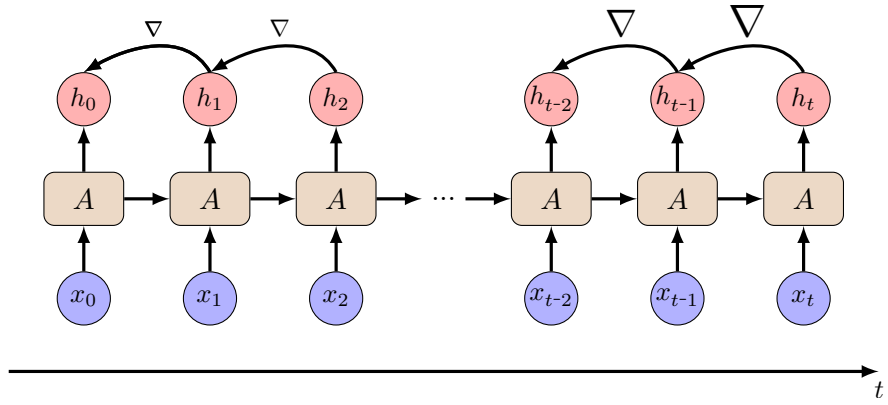
BPTT захтева одређена уопштења, пошто у размотаном графу неурони нису сложени по слојевима у духу мреже са пропагацијом унапред (на пример, излази нису само на последњем нивоу размотаног графа, већ на више различитих нивоа). Међутим, уопштења нису компликована и може се повући паралела у односу на ова два алгоритма.

Дељење параметара код RNN-а и дубина мреже која се производи размотавањем графа не представљају препреку у обучавању, у смислу флексибилности модела, али и даље постоје други проблеми, попут великог броја надовезаних множења извода при рачунању градијента, што лако доводи у неком тренутку до великог смањења (нестајања), а некада и до великог увећања (експлозије) градијената. У циљу бољег разумевања, корисно је

сагледати ефекте BP и BPTT алгоритма на примеру MLP-а и RNN-а, што је илустровано на слици 2.21.



(а) BP



(б) BPTT

Слика 2.21: MLP и RNN у процесу обучавања

Као што је раније представљено, обука неуронске мреже има два главна корака. Први је пролазак унапред, који укључује прављење предвиђања и његово поређење са очекиваним резултатом употребом функције губитка. Добијена вредност грешке представља процену прецизности функционисања мреже. У другом ова вредност се користи за пропагацију уназад и поправљање тежинских веза у мрежи.

Градијент се примењује за прилагођавање унутрашњих тежина у циљу смањења грешке коју мрежа прави, чиме се омогућава њено обучавање. Што је већи градијент, већа су подешавања и обрнуто. Овде лежи проблем. Када се врши пропагација уназад, сваки чвор у слоју израчунава градијент у односу на ефекте промене градијента у слоју пре њега. Дакле, ако је подешавање ранијих слојева мало, онда ће подешавање тренутног слоја бити још мање. То узрокује да се градијенти експоненцијално смањују док пропагација уназад напредује. Плићи слојеви не успевају да уче јер се подешавања унутрашњих слојева једва дешавају због екстремно малих градијената. Дакле, ово је проблем нестајања градијента.

Код RNN архитектуре дешава се слична појава. Формални опис проблема биће приказан у наставку [63].

Ако су параметри RNN модела дати као:

- t - временски корак;
- T - број временских корака;
- W_{rec} - Тежинска матрица рекурентне везе;

- W_{in} - Тежинска матрица улазне везе;
- x_t - улазни вектор у временском кораку t;
- h_t - скривено стање у временском кораку t, где је x_0 унапред задато;
- l_t - грешка у временском кораку t;
- $L = \sum_{t=1}^{T} l_t$ - укупна грешка;
- f - активациона функција;
- θ - обједињени W_{rec}, W_{in} и b.

модел се може описати једначином (2.8).

$$h_t = W_{rec} f\left(h_{t-1}\right) + W_{in} x_t + b \tag{2.8}$$

Једначине (2.9), (2.10) и (2.11) описују BPTT.

$$\frac{\partial L}{\partial \theta} = \sum_{t=1}^{T} \frac{\partial l_t}{\partial \theta} \tag{2.9}$$

$$\frac{\partial l_t}{\partial \theta} = \sum_{k=1}^{t} \left(\frac{\partial l_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial^+ h_t}{\partial \theta} \right) \tag{2.10}$$

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}} = \prod_{i=k+1}^{t} W_{rec}^{\top} \operatorname{diag}\left(f'\left(h_{i-1}\right)\right) \tag{2.11}$$

где $\frac{\partial^+ h_t}{\partial \theta}$, у једначини (2.10), означава непосредни парцијални дериват [74].

Из једначине (2.11) може се закључити да је чинилац W_{rec} одговоран за конвергирање градијента према вредности 0, односно за његово дивергирање. Када важи $W_{rec} < 1$ тада се испољава проблем нестајања, а у случају $W_{rec} > 0$ проблем експлозије градијента. Ови проблеми могу се решавати на више начина.

За проблем експлозије градијента, нека од решења могу бити:

- заустављање BPTT после одређеног броја корака;
- увођење пенала или вештачко редуковање градијента;
- стављање максималног лимита за вредност градијента.

За проблем нестајања градијента нека од решења могу бити:

- иницијализовање тежина тако да могућност нестајања градијента буде минимизована;
- имплементација неуронске мреже са ехо стањем *(енгл. Echo State Network)* [31];
- имплементација LSTM неуронске мреже.

Због нестајања градијента, RNN не учи дугорочне зависности током временских корака. Дакле, немогућност учења у ранијим временским корацима доводи до тога да мрежа има краткорочну меморију.

У прошлој деценији било је значајних успеха у примени RNN-а за решавање разних проблема. Суштина тих успеха била је у примени дуге краткорочне меморије, LSTM-а, посебне врсте рекурентне неуронске мреже која функционише, за већину задатака, много боље од почетних верзија, решавајући проблем краткорочне меморије. Већина значајних резултата, везаних за рекурентне неуронске мреже, постигнута је овом техником.

LSTM успешно задржава W_{rec} у близини вредности 1, а то постиже структурама које се називају капије. Овај механизам биће детаљније представљен у наставку.

2.3 LSTM мреже

2.3.1 Основни принципи LSTM-а

LSTM су посебна врста RNN-а, способна да учи дугорочне зависности. Експлицитно су дизајниране да избегну проблем дугорочних зависности, памћење информације у дугим временским периодима је практично њихова основна особина, а не потешкоћа са којом се боре. Показало се да успешно решавају широк спектар проблема и због тога имају велику примену.

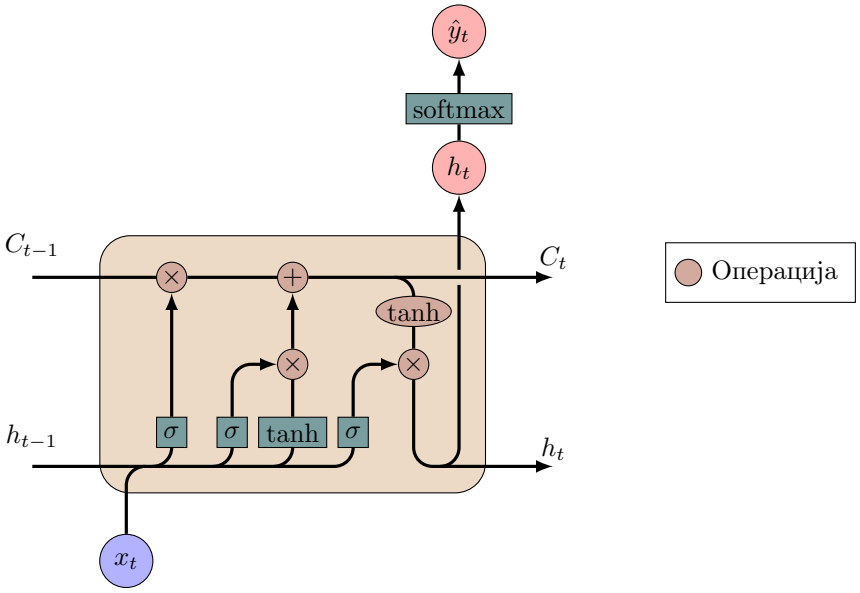
Све рекурентне неуронске мреже имају ланчану форму сачињену од модула (корака) који се понављају. Код стандардних RNN-а, модули ће имати врло једноставну структуру, што је често један слој неурона са одређеном активационом функцијом 2.20. Код LSTM-а, модул који се понавља има другачију конфигурацију и уместо једног неуронског слоја, овде су четири која интерагују на врло специфичан начин [61].

Нека је:

- C - Стање ћелије;
- f - Капија заборављања *(енгл. Forget Gate Layer)*;
- W_f - Тежинска матрица капије заборављања;
- b_f - Померај капије заборављања;
- i - Улазна капија *(енгл. Input Gate Layer)*;
- W_i - Тежинска матрица улазне капије;
- b_i - Померај улазне капије;
- o - Излазна капија *(енгл. Output Gate Layer)*;
- W_o - Тежинска матрица излазне капије;
- b_o - Померај излазне капије;
- $\widetilde{C}$ - кандидат;
- W_C - Тежинска матрица кандидата;
- b_C - Померај кандидата;

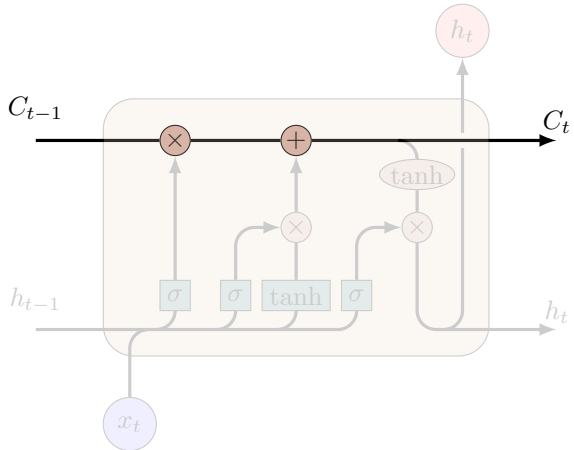
LSTM архитектура може се описати једначинама (2.12) и сликом 2.22.

$$
\begin{aligned}
f_t &= \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \\
i_t &= \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \\
o_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \\
\widetilde{C}_t &= \tanh\left(W_C\left[h_{t-1}, x_t\right] + b_C\right) \\
C_t &= f_t * C_{t-1} + i_t * \widetilde{C}_t \\
h_t &= o_t * \tanh\left(C_t\right) \\
\hat{y}_t &= \operatorname{softmax}\left(W_y h_t + b_y\right)
\end{aligned}
\tag{2.12}
$$



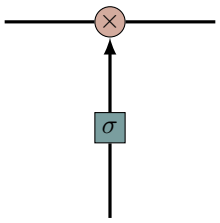
Слика 2.22: Структура LSTM ћелије

Кључ LSTM-а је стање ћелије, хоризонтална линија која пролази кроз врх дијаграма. Стање ћелије понаша се као транспортна трака. Она иде равно низ цео ланац и на њој се појављују одређене линеарне операције, што је илустровано на слици 2.23. Може се десити да информација само тече дуж ње непромењена.



Слика 2.23: Стање ћелије

LSTM има способност да уклони или дода информације у стање ћелије, а то је пажљиво регулисано структурама званим капије. Капије су механизам за селективно пропуштање информација. Оне се састоје од мрежног слоја који користи сигмоид активациону функцију и операцију множења, што је илустровано на слици 2.24.

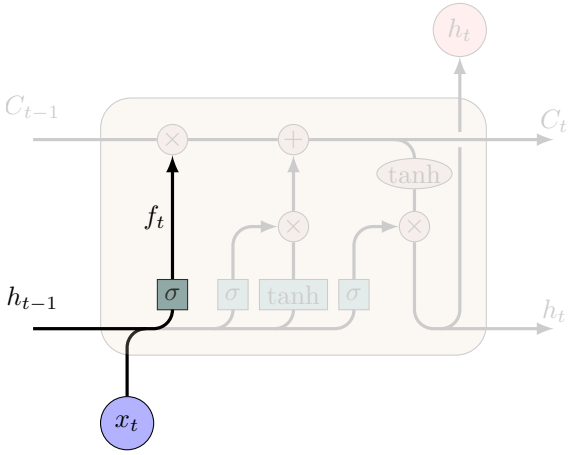


Слика 2.24: Капија

Сигмоидни слој шаље на излаз бројеве из интервала $[0, 1]$, описујући на тај начин колико би информација од сваке компоненте требало проћи. Вредност нула значи не дозволити ништа, док вредност један значи да све пролази. LSTM у основном облику има три овакве капије, како би заштитио и контролисао стање ћелија.

Први корак у LSTM-у је одлука које ће информације бити одбачене из стања ћелије. Ову одлуку доноси први сигмоидни слој познат као капија заборављања. Она прима h_{t-1} и x_t и враћа број из интервала $[0, 1]$ за сваку информацију у стању ћелије C_{t-1}. Јединица представља потпуно задржавање, док нула представља потпуно одбацивање информације. Илустрација се налази на слици 2.25.

Може се направити осврт на пример модела за препознавање природног језика који покушава да предвиди следећу реч на основу свих претходних, из поглавља 2.1.3. У оваквом проблему, стање ћелије може укључити пол садашњег субјекта, да би се могле користити исправне заменице у наредном предвиђању. Када се наиђе на нови субјекат, потребно је заборавити пол старог субјекта.



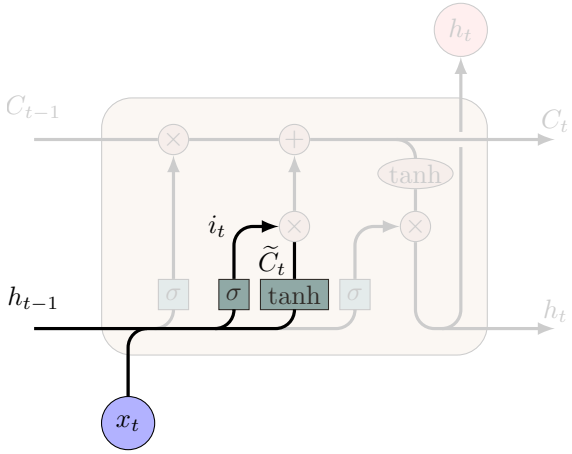
$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right)$$

Слика 2.25: Капија заборављања

Следећи корак је одлука које нове информације треба чувати у стању ћелије. Она се спроводи у два дела. Прво, сигмоидни слој који се назива улазна капија одлучује које вредности треба ажурирати. Затим, у другом делу, слој чија је активациона функција хиперболички тангенс, ствара вектор нових вредности кандидата $\widetilde{C}$, које се могу додати стању. У следећем кораку комбинују се добијене вредности да би се креирало ажурирање за стање, што се може видети на слици 2.26.

У примеру модела за препознавање природног језика, ова капија треба додати пол новог

субјекта у стање ћелије, да би стари био замењен.

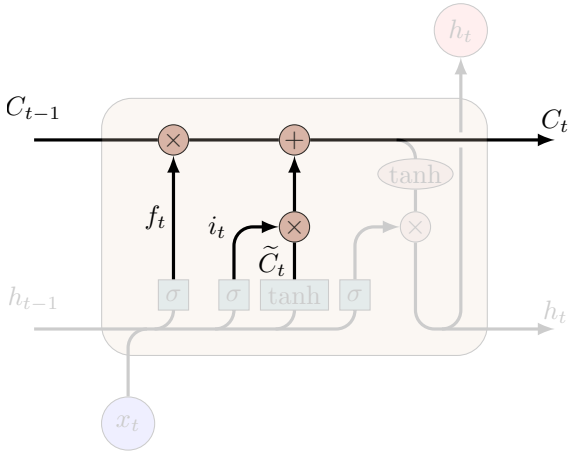


$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right)$$
$$\widetilde{C}_t = \tanh\left(W_C\left[h_{t-1}, x_t\right] + b_C\right)$$

Слика 2.26: Улазна капија

Даље се ажурира старо стање ћелије C_{t-1} чиме настаје ново стање C_t. Претходним корацима је већ у потпуности одређен начин на који ће ажурирање бити извршено.

Множи се претходно стање са f_t чиме се заборављају подаци за које је раније одлучено да их треба заборавити. Затим следи додавање $i_t * \widetilde{C}_t$, што су нове вредности кандидата, скалиране према томе колико је одлучено да треба ажурирати сваку вредност стања. Овај корак илустрован је на слици 2.27.



$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$$

Слика 2.27: Ажурирање стања

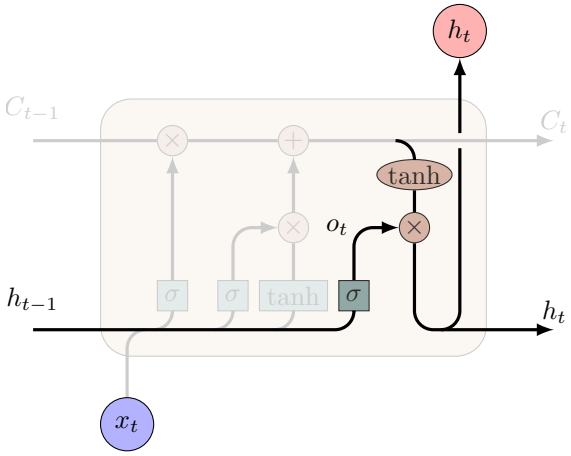
У случају језичког модела, овде би заправо требало одбацити податак о полу старог субјекта и додати нови податак, у складу са оним што је одлучено у претходним корацима.

Коначно, потребно је одлучити шта све од додатака треба проследити на излаз. Излаз ће бити заснован на стању ћелије, али ће притом бити филтриран.

Прво је потребно активирати сигмоидни слој који одлучује који делови стања ћелије су пожељни на излазу. Затим треба пропустити стање ћелије кроз функцију хиперболичког тангенса (што ће резултовати излазом чија је вредност из интервала $(-1, 1)$) и помножити резултат са излазом сигмоидног слоја, тако да буду послати на излаз само делови за које је претходно донета таква одлука. Ово је илустровано на слици 2.28.

Код примера модела за препознавање природног језика, након што је виђен субјекат, може бити пожељно да се изведу закључци релевантни за глагол, у случају да је глагол оно што следи. На пример, на излазу је потребно дати информацију о томе да ли је субјекат у

једнини или множини, и тако добити сазнање у коју форму би актуелни глагол требало да се конјугује.



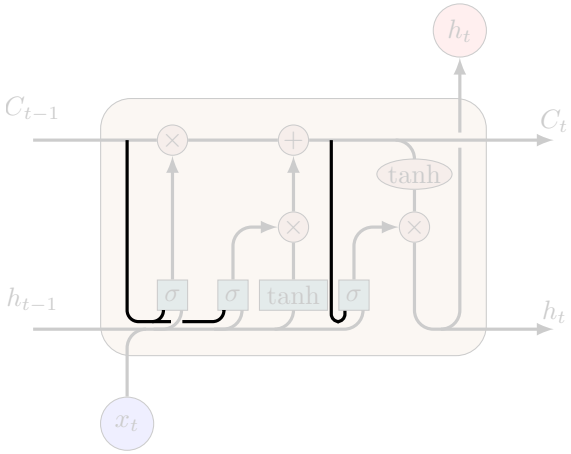
$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Слика 2.28: Излазна капија

2.3.2 Варијанте LSTM-а

До сада је било речи о стандардним LSTM. Међутим, постоји пуно различитих верзија. Разлике су мале, али вреди поменути неке од њих.

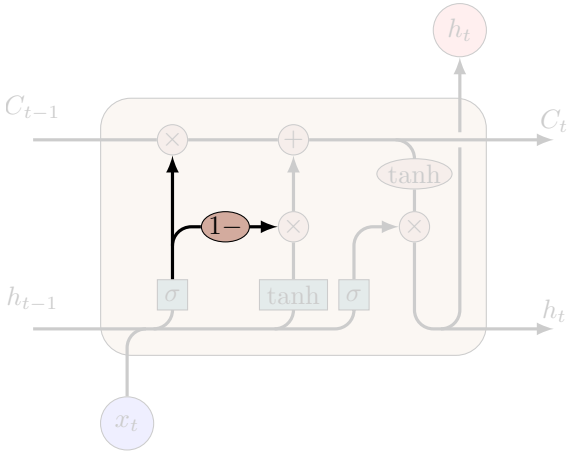
Једна популарна LSTM варијанта, коју су увели 2000. године Герс *(нем. Felix Gers)* и Шмитхубер [21], додаје везе за шпијунирање *(енгл. peephole)* које би требало да помогну у прецизнијем учењу информација. То значи да је дозвољено да слојеви капија имају увид у стање ћелије. Дијаграм на слици 2.29 додаје *peephole* свим капијама, али постоје и варијанте које то раде селективно.



$$f_t = \sigma\left(W_f\left[C_{t-1}, h_{t-1}, x_t\right] + b_f\right)$$
$$i_t = \sigma\left(W_i\left[C_{t-1}, h_{t-1}, x_t\right] + b_i\right)$$
$$o_t = \sigma\left(W_o\left[C_t, h_{t-1}, x_t\right] + b_o\right)$$

Слика 2.29: Везе за шпијунирање

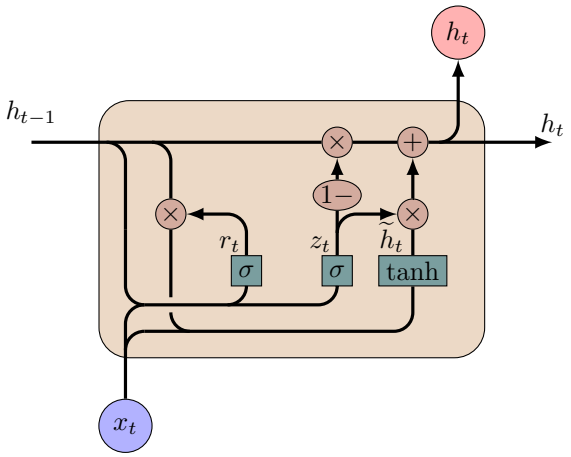
Постоји и врста LSTM-а која користи капију заборављања у спрези са улазном капијом. Уместо да се одвојено доноси одлука о томе које информације треба одбацити и где треба додати нове, те одлуке се доносе истовремено. Одбацује се само податак који је потребно заменити новим. Нови подаци се на тај начин памте, само када се одбацују њихове старије верзије. Овај приступ илустрован је на слици 2.30.



$$C_t = f_t * C_{t-1} + (1 - f_t) * \widetilde{C}_t$$

Слика 2.30: Комбиновани улаз и заборављање

Радикалније измењена верзија LSTM-а је рекурентна јединица ограничена капијама *(енгл. Gated Recurrent Unit – GRU)*, увео ју је 2014. године Чо *(енгл. Kyunghyun Cho)* [14]. Она комбинује улазну и капију заборављања у једну капију ажурирања *(енгл. Update Gate)*. Она такође спаја стање ћелије и скривено стање и уводи још неке промене. Добијени модел је једноставнији од стандардних LSTM модела и захваљујући томе стекао је популарност. Дијаграм GRU ћелије дат је на слици 2.31.



$$z_t = \sigma\left(W_z\left[h_{t-1}, x_t\right] + b_z\right)$$
$$r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right] + b_r\right)$$
$$\widetilde{h}_t = \tanh\left(W_h\left[r_t * h_{t-1}, x_t\right] + b_h\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t$$

Слика 2.31: GRU

Представљене су само неке од најзначајнијих LSTM варијанти. Много је других, као што су рекурентне мреже дубоко ограничене капијама *(енгл. Depth Gated RNN)*, које је увео 2015. године Јао *(енгл. Kaisheng Yao)* [92]. Постоји и сасвим другачији приступ решавању дугорочних зависности, као што су сатни RNN *(енгл. Clockwork RNN)*, који је увео Кутник *(чеш. Jan Koutník)* 2014. године [41].

Греф *(нем. Klaus Greff)* [23] је 2015. године направио поређење популарних варијанти LSTM-а, открившши да су све скоро подједнако учинковите. Џозефович *(енгл. Rafal Jozefowicz)* [33] је исте године тестирао више од десет хиљада различитих RNN архитектура, пронашавши неке које су радиле боље од LSTM-ова за одређене задатке.

Глава 3

Машинско превођење

Машинско превођење *(енгл. Machine Translation – MT)* је посебна област компјутерске лингвистике која истражује употребу софтвера у превођењу текста или говора са једног природног језика на други.

Употреба механичких уређаја за превазилажење језичких баријера предложена је још у XVII веку. Због пропасти латинског, као универзалног језика за научну комуникацију, и тада претпостављене неадекватности природних језика за изражавање истраживачких размишљања, настала је идеја универзалних језика из жеље да се побољша међународна комуникација и створи рационално и логично средство научне комуникације. Предлози за нумеричке кодове који би посредовали међу језицима представљали су почетну идеју. Лајбницови *(нем. Gottfried W. von Leibniz)* предлози у контексту његове монадичке теорије можда су и најпознатији. Декарт је предлагао универзални језик у облику шифре где би лексички еквиваленти свих познатих језика добили исти кодни број [29].

Први покушаји да се направи уређај који би могао аутоматски да преводи почели су тридесетих година прошлог века, али из више разлога први конкретни резултати појавили су се тек 1954. године. У оквиру *Georgetown-IBM* експеримента створен је уређај који је преводио са руског на енглески. Овај апарат ослањао се на фонд од само 250 речи, као и на чињеницу да су реченице из научног текста, које су превођене током демонстрације, биране унапред од стране твораца система због једноставности синтаксе [40].

Током шездесетих година, влада САД-а, незадовољна дотадашњим напретком, укинула је финансирање пројеката везаних за машинско превођење. Једна од ретких компанија која је преживела тај период, приватно финансирани *Systran*, наставио је развој система започетог на Универзитету Џорџтаун [40].

Следећи велики корак десио се тек осамдесетих година, када је пораст доступности рачунара утицао на даљи напредак система за превођење. До средине деведесетих година водећи ауторитети на пољу машинског превођења били су *IBM*, *Logos* и *Systran*. На *Systran* машинском преводиоцу био је заснован и *Babel Fish*, први широко познати бесплатни систем за превођење, који је заживео почетком 1999. године. Четири године по настанку, *Babel Fish* преузела је *AltaVista*, која је дуже времена затим била део компаније *Yahoo*. Данас доминантну улогу у машинском превођењу има компанија *Google* [40].

Машинско превођење данас карактерише велика брзина самог процеса и ниска цена. Главна мана је још увек недовољно добар квалитет превода за примену у неким областима, али и поред тога у разним ситуацијама може олакшати живот, а преводиоцима послужити као добар алат за убрзавање поступка превођења.

3.1 Приступ

3.1.1 Машинско превођење базирано на правилима

Први приступи у изградњи система за машинско превођење заснивали су се на строгим лингвистичким правилима *(енгл. Rule-based machine translation – RBMT).*

На најнижем нивоу МТ једноставно преводи реч по реч, замењујући речи једног језика одговарајућим речима другог језика. Овај принцип може бити користан у неким доменима који примењују веома ограничен и формализован језик, као што је случај са временском прогнозом. Међутим, за добар превод текстова, који нису толико стандардизовани, треба већим текстуалним јединицама, као што су фразе, реченице или пасуси, наћи одговарајуће еквиваленте у циљаном језику [89].

Највећа потешкоћа са којом се суочава овај приступ је вишезначност природних језика. То ствара изазове на различитим нивоима. Треба, на пример, отклонити вишезначност речи на лексичком нивоу:

Коса може бити име, део тела, придев и алатка;

или утврдити повезаност предлошких фраза на синтаксичком нивоу:

Полицајац је приметио човека *без двогледа.*
Полицајац је приметио човека *без револвера.*

За превођење између сродних језика могућ је и директан приступ у случајевима који наликују наведеним примерима. Ипак, напреднији RBMT систем анализира улазни текст и креира посредну симболичку интерпретацију из које се потом генерише текст на циљаном језику. Успешност оваквог приступа веома зависи од постојања исцрпних лексикона са морфолошким, синтаксичким и семантичким подацима и великих скупова граматичких правила пажљиво креираних од стране искусних лингвиста. Све то је веома дуг и скуп процес.

Први значајнији успеси везани за RBMT појавили су се 70-их година, а најпознатији представник је *Systran.*

Предности овог приступа су:

- Двојезични текстови нису потребни.
- Независност од домена. Правила се обично пишу независно од врсте текста, тако да ће велика већина правила важити у сваком домену.
- Нема ограничења у односу на квалитет текста. Свака грешка може се исправити циљаним правилом, чак и ако је изузетно ретка. То је у супротности са начином на који раде статистички системи за превођење, где ће ретки обрасци подразумевано бити одбачени.
- Тотална контрола. Будући да су сва правила ручно написана, лако се могу отклонити грешке и тачно видети где дата грешка улази у систем и зашто.
- Поновна употреба. RBMT системи детаљно анализирају изворни језик, резултати анализе се преносе кораком преноса, и на крају генерише се превод на циљаном језику. Анализа изворног језика и делови генерисања превода могу се делити између више система превођења, захтевајући да се специјализује само корак преноса. Поред тога, анализа једног језика може се поново користити као основа за анализу другог сродног језика.

Мане овог приступа су:

- Недостатак адекватних речника. Изградња нових речника је скупа.
- Неке језичке информације и даље треба поставити ручно.
- Тешко је бавити се интеракцијама правила у великим системима, двосмисленостима и идиоматичним изразима.
- Тешко прилагођавање. Иако RBMT системи обично пружају механизам за креирање нових правила и проширивање и прилагођавање лексикона, промене су обично веома скупе, а резултати често незадовољавајући.

3.1.2 Машинско превођење базирано на примерима

Напреднији приступ MT-у појавио се у Јапану осамдесетих година и заснован је на постојећим примерима превода *(енгл. Example-based Machine Translation – EBMT)*. Јапан се посебно интересовао за машински превод. Постојала су два битна разлога за то:

- Врло мало људи у земљи је знало енглески. То је наговештавало пуно проблема у предстојећој фази глобализације. Дакле, Јапанци су били изузетно мотивисани да пронађу ефикасну методу машинског превођења.
- Енглеско-јапански превод заснован на правилима је изузетно компликован. Језичка структура је потпуно другачија, при преводу готово све речи морају се преуредити и додати нове.

Нагао *(енгл. Makoto Nagao)*, са Универзитета у Кјоту, 1984. године дошао је на идеју да користи готове изразе уместо да целокупно превођење врши од почетка [58].

Човек не преводи једноставну реченицу радећи дубоку лингвистичку анализу, већ то ради, исправним декомпоновањем дате реченице у одређене фрагментарне фразе, а затим, правилним састављањем ових фрагментарних превода у реченицу. Свака фрагментарна фраза преводи се по принципу аналогне транслације са датим одговарајућим примерима као референцом. – М. Нагао

Дат је пример једноставне реченице коју треба превести - *Идем у биоскоп*. А је већ преведена слична реченица - *Идем у позориште*. У речнику се може наћи реч *биоскоп*. Све што је даље потребно јесте схватити разлику између две реченице, превести реч која недостаје, а затим је исправно додати. Што више преведних примера постоји, превод је бољи.

EBMT је указао на потпуно другачији приступ: испоставило се да је машину могуће само нахранити постојећим преводима и не трошити године на правила и изузетке. Овај приступ ипак није постигао завидне резултате и не може се сматрати револуционарним, али је први корак ка револуцији.

Предности овог приступа су:

- Кореспонденције се могу закључити из сирових података.
- Примери дају добро структуриран излаз ако је подударање довољно велико.

Мане овог приступа су:

- Недостатак добро поравнатих вишејезичних превода.
- Генерисани текст обично је некохезиван.

3.1.3 Статистичко машинско превођење

Почетком 1990. године у *IBM* истраживачком центру први пут је приказан систем машинског превођења који се није у потпуности базирао на правилима и лингвистици, већ је користио статистички приступ *(енгл. Statistical Machine Translation – SMT)*. Систем је анализирао сличне текстове на два језика и покушавао да разуме обрасце [10].

Статистички машински превод није могућ без веома великог броја одобрених превода, који се користе за аутоматско обучавање SMT модела. Обучени модел се затим примењује на непреведеним текстовима и на основу научене вероватноће, предлаже превод.

У SMT-у, модел превођења се гради коришћењем фреквенције фраза које се појављују у корпусу. У табели се чува фраза и број њених понављања. Што се чешће преведена фраза понавља у корпусу за тренирање, већа је вероватноћа да је превод тачан. Свака фраза (сачувана у табели фраза) може имати дужину од једне до, најчешће, не више од пет речи. Оваква конструкција назива се n-грам модел.

SMT модел је еволутиван, корпус се усавршава и прилагођава након сваког покретања превођења како би се уклониле или прилагодиле евентуалне аномалије. Што се корпус више проширује, то модел постаје бољи. Развој корпуса је континуирани истраживачки процес, веома драгоцен за MT.

Ова метода показала се као много ефикаснија и прецизнија од свих претходних. Од 2007. до 2016. године сервис *Google Translate* користио је овај приступ.

Предности овог приступа су:

- Велика предност SMT-а је широка доступност различитих релевантних платформи и алгоритама. То као резултат, омогућава брзо тренирање и додавање нових језика у поређењу са другим MT моделима.
- SMT захтева мање виртуелног простора од осталих модела, што олакшава рад и обуку на мањим системима.
- Добро обучен, прилагођен корпус може доследно преводити свеобухватни садржај, међутим, превод често садржи грешке које захтевају накнадно уређивање.

Мане овог приступа су:

- Превођење текста који садржајно није сличан корпусу на коме је модел трениран. SMT може постићи добре резултате када се примени на материјалу чији је домен дефинисан тренинг корпусом. На пример, ако је тренинг вршен над скупом техничких текстова који су написани једноставним стилом, постојаће потешкоће при превођењу текста који садржи сленг или идиоме. У таквим случајевима тачност SMT-а опада, што условљава одређивање модела према датом стилу. Чак и тада, SMT није у могућности да преведе идиоме и маркетиншки материјал, а употреба код слободног стила резултира лошом тачношћу.
- SMT системима је за тренинг потребна велика количина двојезичних садржаја и то може бити проблем када су у питању ретки језици.
- SMT може бити скуп, иако је много јефтинији од већине других метода. Припремање и стварање корпуса су најзахтевнији у том погледу.
- Теже је исправити грешке у систему након што су имплементиране. Помоћу модела попут RBMT-а могу се исправити грешке и прилично лако уклонити одређене речи. Код SMT-а мора се поново обучити цео систем и потом проверити да ли су настали други проблеми.

3.1.4 Неуронско машинско превођење

Неуронски машински превод *(енгл. Neural Machine Translation – NMT)* је нови приступ машинском превођењу који користи вештачку неуронску мрежу да предвиди вероватноћу секвенце речи, обично преводећи читаве реченице у једном интегрисаном моделу.

Идеја коришћења неуронске мреже у MT-у први пут се појавила у истаживању које је спровео Чо 2014. године [14]. Већ 2016. године, компанија *Google* је објавила да сервис *Google Translate* почиње да користи NMT приступ.

За две године, неуронске мреже надмашиле су све друге приступе који су се користили и непрекидно развијали претходних деценија у сфери MT-а. NMT је када се појавио садржао 50% мање грешака у редоследу речи, 17% мање лексичких грешака и 19% мање граматичких грешака. Неуронске мреже су успешно научиле да хармонизују род и падеж у различитим језицима, само на основу датих примера [37].

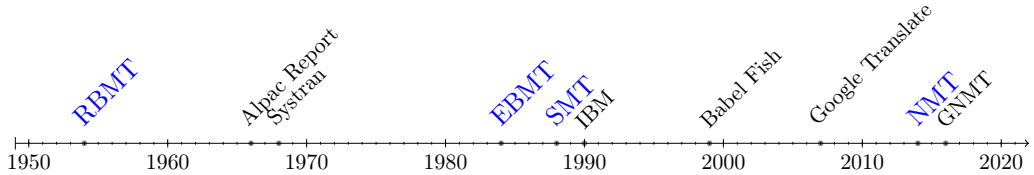
Најуочљивија побољшања догодила су се у пољима где директни превод никада није коришћен. Статистичке методе машинског превођења углавном су радиле тако што су користиле енглески језик као кључни извор. На пример, ако се преводи са руског на немачки, SMT текст преводи прво на енглески, а затим са енглеског на немачки, што доводи до двоструког губитка. NMT-у тај корак није потребан. Први пут директно превођење између језика, без заједничког речника, постало је могуће.

Предности овог приступа су:

- Већи квалитет превода у односу на друге методе.
- Мање потребних података за тренирање у односу на SMT.
- Директно превођење.
- Толеранција на податке са грешкама.
- Јефтинији развој.

Мане овог приступа су:

- Потребно је пуно времена за обучавање.
- Потребно је више времена за превођење у односу на SMT.
- Захтева значајне хардверске ресурсе.
- Истренирани модел није подложан интерпретацији.



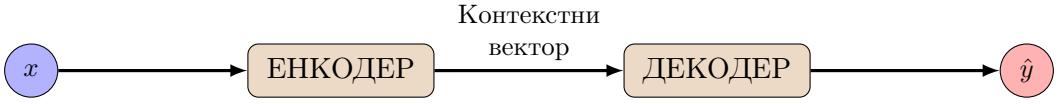
Слика 3.1: Значајни догађаји у развоју MT-а

3.2 Енкодер-декодер модел

3.2.1 Архитектура

Поред проблема дугорочних зависности, постоји још један битан изазов у језичким моделима који користе неуронске мреже, а то је проблем варијабилних дужина улаза и излаза. Енкодер-декодер *(енгл. Encoder-Decoder – ED)* модели, познати још као секвенца-у-секвенцу *(енгл. Sequence-to-Sequence)* модели, показали су способност да реше овај проблем. ED може се окарактерисати као генеративни NN модел који, с обзиром на низ улаза, производи низ резултата, где су оба низа произвољне дужине. Први пут ова архитектура представљена је од стране компаније *Google* 2014. године [79].

На најосновнијем нивоу, ова архитектура функционише тако што енкодер прихвата реченицу на једном језику и ствара мисаони *(енгл. Thought Vector)* или контекстни *(енгл. Context Vector)* вектор на основу ње. Овај вектор представља значење реченице, које се у наредном кораку прослеђује декодеру који на основу њега даје превод на другом језику. Овај поступак је приказан на слици 3.2.



Слика 3.2: Енкодер-декодер архитектура

У приказаној архитектури енкодер и декодер су рекурентне неуронске мреже. Углавном су у питању LSTM мреже, често се користи и GRU, док су и друге RNN имплементације могуће.

NMT, као релативно нов приступ, константно се унапређује, а истраживања у овој области последњих година интензивирају. У наставку биће речи о техникама које се користе за изградњу ED модела.

3.2.2 One Hot кодирање

Код MT-а улаз представљају реченице на једном језику, а задатак је превођење истих на други језик. Једна од очигледних препрека у том процесу је питање руковања текстуалним подацима, с обзиром да неуронске мреже природно баратају бројевима.

One hot кодирање је једноставно решење поменутог проблема које претвара све речи у скупу података у векторе нула и јединица. Ти вектори имају дужину која одговара броју свих речи у речнику који се користи за одређени задатак. Реч се кодира тако што се свака позиција у вектору попуни нулом, са изузетком једне позиције која одговара тој одређеној речи, која се попуњава јединицом. Да би била одређена величина вектора за кодирање, креира се засебан вокабулар за оба језика.

Идеално би било да речници садрже све речи оба језика. Међутим, с обзиром на то да један језик може имати милионе јединствених речи, речници се често састоје од подскупова најчешћих речи оба језика. Речницима се додају и посебне речи као што су $< SOS >$ која означава почетак и $< EOS >$ која означава крај реченице. Могуће је из више разлога уводити и друге специјалне речи као што је $< UNK >$, која представља непознату реч. Специјалне речи постају важне током тренажног процеса и о њима ће бити речи у наставку.

One hot репрезентација не садржи никакве информације о кодираним речима и њиховом међусобном односу, али има јако погодно својство да захтева малу количину меморије потребне за једнозначно одређивање речи, за шта је довољан само индекс дате речи у речнику.

За дати речник у тебели 3.1 дат је пример кодирања реченице на слици 3.3.

Реч	Индекс
Ана	0
Аца	1
воли	2
да	3
једе	4
јабуке	5
крушка	6
.	7
$< SOS >$	8
$< EOS >$	9

Табела 3.1: Пример речника

$$Аца = \begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} \; воли = \begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} \; да = \begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix} \; једе = \begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\\0\\0\end{bmatrix} \; јабуке = \begin{bmatrix}0\\0\\0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix} \; . = \begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\1\\0\\0\end{bmatrix}$$

Слика 3.3: One hot кодирана реченица: *Аца воли да једе јабуке.*

3.2.3 Word Embedding

One hot кодирање је заправо једноставна репрезентација у којој се свака реч пресликава у јединствени вектор. Ова техника има два главна недостатка:

- Кодирање скупова велике кардиналности производи векторе великих димензија.
- Добијени вектор не садржи информације о кодираној речи.

Први проблем условљава додавање још једне димензије вектору који представља код, за сваку нову реч додату у речник. Овај проблем обуку било ког модела са речником од пар десетина хиљада речи чини у пракси јако тешко изводљивом.

Други проблем је такође веома ограничавајући: One hot кодирање не издваја сличне категорије у векторском простору. Мерењем сличности између вектора помоћу косинусне удаљености, добија се сличност 0 за било који пар речи. То проузрокује да модел не пропознаје сличне речи, што је у задацима попут МТ-а јако значајно.

Word embedding представља класу техника којима се речи репрезентују као вектори реалних вредности у унапред дефинисаном векторском простору. Свака реч се пресликава у посебан вектор чије се вредности добијају тренирањем над корпусом. Кључна идеја за овај приступ је употреба густих вектора *(енгл. Dense Vector)* за репрезентацију речи. Свака реч представљена је вектором који садржи десетине или стотине димензија. То је велико побољшање у поређењу са милионима димензија потребних за ретке векторе *(енгл. Sparse Vector)*, на којима се заснива one hot кодирање.

Репрезентација се учи на основу употребе речи у корпусу. То омогућава да речи које се користе на сличне начине буду слично приказане. Иза приступа стоји дубља лингвистичка теорија и хипотеза о дистрибуцији коју је изнео Харис *(енгл. Zellig S. Harris)* [25], а која би се могла сажети у реченици:

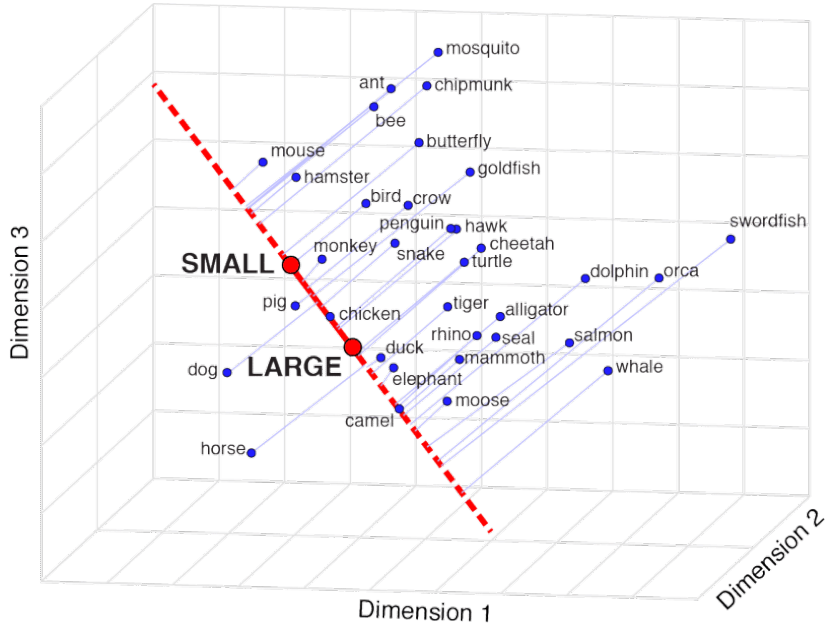
Речи сличног контекста имаће слична значења.

Word embedding учи на основу метода које стварају векторски приказ речи на основу предефинисаног вокабулара фиксне величине и корпуса текста. Биће поменуте три често примењиване технике:

- Embedding слој је додатни мрежни слој који се обучава заједно са моделом. Захтева пречишћен и припремљен текст у коме је свака реч јединствено кодирана. Обично се припрема врши one hot кодирањем. Димензионалност векторског простора одређена је као хиперпараметар модела и варира у распону од пар стотина до пар хиљада. Вектори се иницијализују малим случајним бројевима. Embedding слој користи се као први слој неуронске мреже и обучава на надгледан начин коришћењем BP алгоритма.
- Word2Vec је статистичка метода за ефикасно учење самосталног word embedding-а на основу текстуалног корпуса. Развила га је компанија *Google* 2013. године у циљу побољшања ефикасности тренирања модела заснованих на неуронским мрежама и од тада је постао стандард за развој унапред обученог word embedding-а [56]. Поред тога, рад је укључивао анализу научених вектора и истраживање векторске математике на репрезентацијама речи. Као резултат тога, на пример, одузимање речи *мушкарац* од речи *краљ* и додавање речи *жена* резултује речју *краљица*.
- GloVe *(енгл. The Global Vectors for Word Representation)* је алгоритам који представља проширење Word2Vec метода. Резвијен на Универзитету Стенфорд 2014. године [64]. Комбинује више математичких метода анализе корпуса и као резултат даје модел који генерално има боље перформансе од Word2Vec-а.

Тренирање word embedding-а заједно са моделом показује генерално најбоље резултате, али са друге стране повећава време тренирања модела. Word2Vec и GloVe су бесплатни за преузимање у више варијанти које се односе на димензионалност резултујућег вектора. Такође, код њих постоји могућност додатног тренирања, како би се што боље прилагодили моделу у коме се користе.

Word embedding прави битну прекретницу у обради природних језика. Његова примена знатно побољшава перформансе језичких модела. На слици 3.4 приказана је 3D пројекција репрезентације речи у векторском простору и пример њихове међусобне релације.



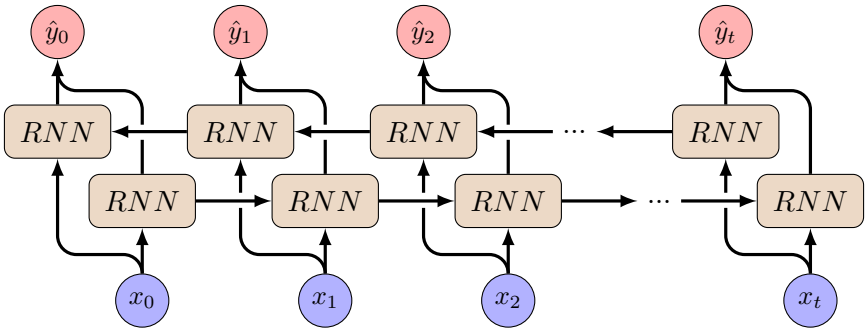
Слика 3.4: Word embedding 3D пројекција [22]

3.2.4 Двосмерни LSTM

Двосмерне RNN *(енгл. Bidirectional RNN - BRNN)* настају повезивањем два скривена рекурентна слоја супротних праваца на исти излаз. Помоћу овог облика дубоког учења, излазни слој може истовремено добити информације из претходних и будућих стања. BRNN су уведене 1997. године од стране Шустера *(енгл. Mike Schuster)* и Паливала *(енгл. Kuldip Paliwal)* са циљем увећавања количине улазних информација доступних мрежи [77].

Двосмерни LSTM-ови *(енгл. Bidirectional LSTM - BLSTM)* усредсређени су на проблем извлачења максимума информација из улазног низа проласком кроз временске кораке уноса у оба смера. Двосмерни приступ је показао одличне ефекте управо са овом врстом RNN-а у задацима као што су: препознавање говора и машинско превођење.

Употреба двосмерних LSTM-ова нема смисла за све проблеме предвиђања секвенци, али може понудити одређене бенефите у смислу бољих резултата у оним доменима где је то прикладно. У проблемима где су доступни сви временски кораци секвенце уноса, двосмерни приступ тренира два уместо једног LSTM-а над улазним низом. Први на улазној секвенци каква јесте и други на обрнутој копији улазног низа. То може пружити додатни контекст мрежи и резултовати бржим и потпунијим учењем проблема. На слици 3.5 дата је илустрација BRNN мреже.



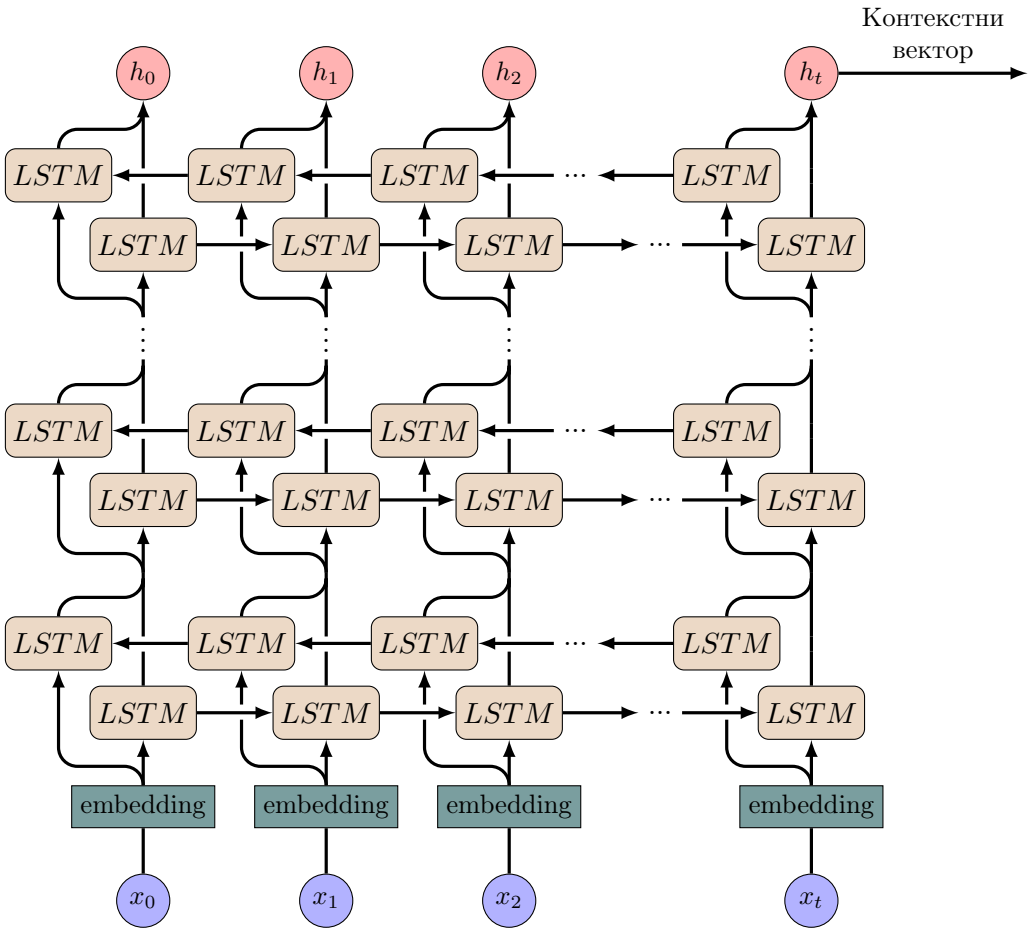
Слика 3.5: BRNN

3.2.5 Енкодер

Енкодер је RNN која има задатак да сваку реч улазне реченице уноси у модел засебно, у више узастопних временских корака. У сваком временском кораку енкодер ажурира скривено стање, користећи информацију коју носи реч унесена у тренутном временском кораку. Скривено стање RNN-а, током времена, акумулира информације о целој унесеној реченици. На почетку оно је репрезентовано празним вектором, који сваком новом речју постаје садржајнији.

У сваком временском кораку скривено стање преузима информације из унете речи, чувајући акумулиране податке из претходних временских корака. У завршном временском кораку, значење целе улазне реченице бива сачувано у вектору који репрезентује скривено стање. Тај вектор назива се контекстни вектор и даље се прослеђује декодеру.

Сама структура енкодера састоји се од једног или више слојева RNN-а. За задатак NMT-а најчешће су у употреби LSTM мреже, где због бољег извлачења информација из улаза може бити примењена њихова BLSTM варијанта. Word embedding је, такође, битна компонента енкодера и његова употреба знатно побољшава перформансе. На слици 3.6 приказана је вишеслојна BLSTM архитектура енкодера.



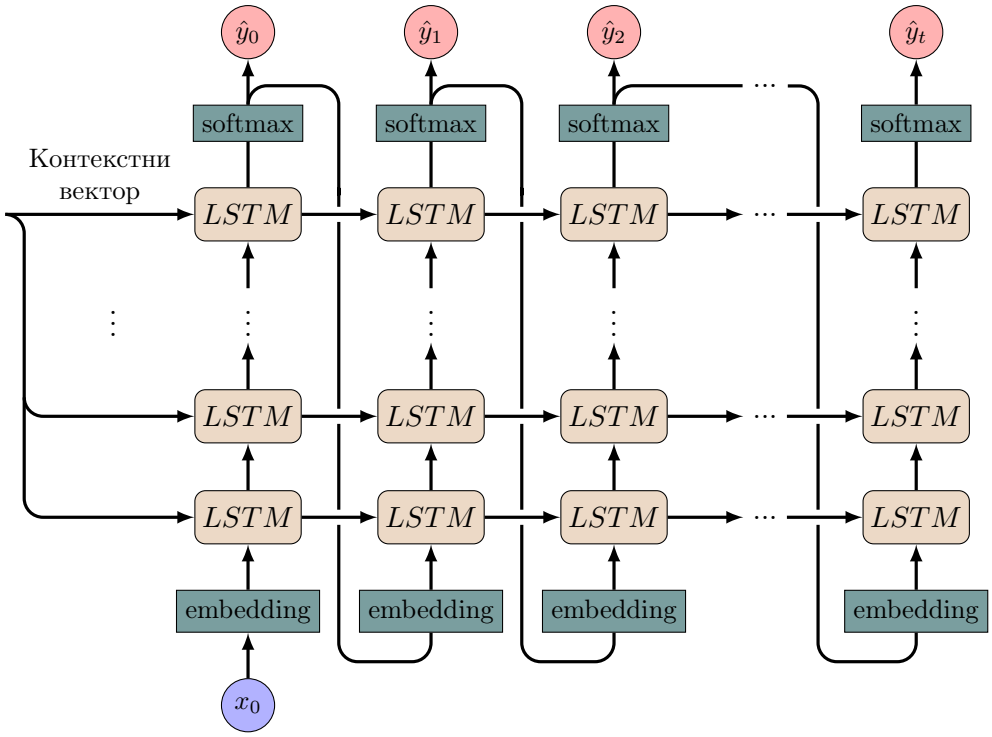
Слика 3.6: Вишеслојни BLSTM енкодер

3.2.6 Декодер

Коначно скривено стање енкодера постаје контекстни вектор, а уједно и иницијално скривено стање другог главног дела NMT неуронске мреже - декодера. Његова улога је преузимање кодираног значења и превођење истог у реченицу на циљаном језику.

Декодер има задатак да на излазу генерише реченицу промењиве дужине. Сходно томе, он ће у сваком временском кораку генерисати реч све док не преведе целу реченицу. За разлику од енкодера, који одмах прелази на следећи временски корак по ажурирању скривеног стања, декодер на крају сваког временског корака користи додатни неуронски слој да би одредио вероватноћу појављивања у датом моменту за све речи у излазном речнику. На овај начин, употребом softmax слоја, реч са највећом вероватноћом постаће следећа реч у предвиђеној излазној реченици. Добијена реч се шаље као улаз у наредном кораку, да би се и она узела у обзир у следећој итерацији.

Архитектура декодера углавном се састоји од више RNN слојева, а LSTM је најчешће имплементирани тип. Word embedding, као и у случају енкодера, побољшава могућности тумачења речи, а самим тим и укупне перформансе модела. На слици 3.7 илустрована је архитектура декодера.



Слика 3.7: LSTM декодер

3.2.7 Специјалне речи

Енкодер-декодер модел је структуриран тако да енкодер прво чита улазну секвенцу да би конструисао контекстни вектор. Пошто су улази промењиве дужине, потребно је направити механизам који ће сигнализирати завршно стање.

Елегантан начин да се то постигне је увођење специјалне речи $< EOS >$ *(енгл. End of Sequence)* и њено постављање на крај сваке реченице тренинг скупа. Тако се даје сигнал енкодеру да када прими тај улаз, излаз мора бити контекстни вектор. Пошто међустања у енкодирању немају употребну вредност ван самог енкодера, ово представља једноставан и ефикасан начин за препознавање краја варијабилне секвенце.

Реч $< EOS >$ је важна и за декодер: експлицитна реч за крај реченице омогућава декодеру да емитује низове произвољне дужине. Декодер овом речју шаље обавештење када се предвиди крај реченице. Без речи $< EOS >$, не би се могло једноставно утврдити када декодер треба престати са даљим предвиђем речи.

Специјална реч за почетак секвенце је $< SOS >$ *(енгл. Start of Sequence)* и важна је за декодер. Декодер ће напредовати узимајући речи које емитује као излазе (заједно са скривеним стањем), тако да пре него што је емитовано било шта, за почетак му је потребна нека реч, која је представљена са x_0 на слици 3.7. Тако је уведена реч $< SOS >$.

Поред тога, ако енкодер користи двосмерни RNN, обавезно се морају увести и $< SOS >$ и $< EOS >$ токени, јер ће $< SOS >$ токен у том случају сигнализирати реверзном слоју када је његов улаз завршен.

Још једна специјална реч која се може користити је $< UNK >$ *(енгл. Unknown)* која представља непознату реч. Заменом нискофреквентних речи у корпусу са $< UNK >$, NMT модел стиче веома битну способност да барата и са речима са којима се није сусретао током процеса обуке.

3.2.8 Attention механизам

Можда и најзахтевнији аспект NMT-а је руковање дугим улазним текстом. С обзиром на природу временских секвенци RNN структура које се користе у машинском превођењу, што је дужа улазна реченица, то је више корака потребно енкодеру да створи контекстни вектор. Даље, што је дужа улазна реченица, више информација је потребно похранити у тај вектор, који потом декодер треба да протумачи.

Код већине реченица које треба превести, на пример, са енглеског на француски језик, редослед речи у оригиналу и преводу углавном је исти. Међутим, чак и у овом случају, како реченица постаје дужа, енкодеру постаје тешко да сачува информације са њеног почетка у контекстном вектору који шаље декодеру. Ако се даље декодеру пружи недовољно информација о томе како треба започети излазну реченицу, врло вероватно резултат неће бити задовољавајући.

У циљу борбе са овим проблемом, раније је уочено да се реверзним уносом улазне реченице, осигурава да речи које одговарају првим речима улазне реченице буду прослеђене у последњим временским корацима енкодеру, тиме осигуравајући да информације о њима буду кодиране у контекстни вектор и прослеђене декодеру. Међутим, иако ова метода има одређеног успеха у превођењу између неких језика, она не генерише добар превод између свих језика (с обзиром на то да структура реченица код различитих језика знатно варира). Даље, ова метода жртвује знатну количину информација са краја улазне реченице, јер се те информације прве уносе и често им се не може приступити при генерисању краја излазне реченице, што резултује лошим преводом после одређеног корака.

Боље решење проблема превођења дугих реченица је attention механизам, који је увео Бахданау *(енгл. Dzmitry Bahdanau)* 2014. године и метод је постао стандард у модерним NMT архитектурама [5].

Луонгова *(енгл. Minh-Thang Luong)* генерализација attention механизама [50] користи скривена стања која емитује у сваком кораку, што омогућава декодеру да се током читавог процеса не ослања само на контекстни вектор. На тај начин декодер се може упутити на одређени део изворне реченице у сваком кораку како би утврдио који њени делови су од највеће помоћи у одређивању следеће речи. Уопштено, attention механизам даје веће тежине скривеним стањима енкодера која се више односе на тренутно скривено стање декодера. То се врши рачунањем контекстног вектора у сваком временском кораку декодера, као пондерисаног просека над скривеним стањима енкодера.

Пондерисани просек одређује се на основу вектора поравнања a_t чија дужина одговара броју скривених стања енкодера. Овај вектор се добија поређењем сличности тренутног скривеног стања декодера h_t^D и сваког засебног скривеног стања енкодера $h_{t'}^E$ које се узима у обзир. На овај начин, вектор a_t дефинисан је једначином (3.1).

$$a_{t_{t'}} = \frac{e^{score\left(h_t^D, h_{t'}^E\right)}}{\sum_{\tau=0}^{K-1} e^{score\left(h_t^D, h_\tau^E\right)}} \qquad \forall t' \in [0, K-1] \tag{3.1}$$

K представља број речи у улазној реченици, а функција *score* одређује сличност између датих скривених стања и може имати више различитих форми које су дате једначинама (3.2), (3.3) и (3.4).

$$score\left(h_t^D, h_{t'}^E\right) = {h_t^D}^\top h_{t'}^E \qquad (dot) \tag{3.2}$$

$$score\left(h_t^D, h_{t'}^E\right) = {h_t^D}^\top W_a h_{t'}^E \qquad (general) \tag{3.3}$$

$$score\left(h_t^D, h_{t'}^E\right) = V_a^\top \tanh\left(W_a \left[h_t^D : h_{t'}^E\right]\right) \qquad (concatenation) \tag{3.4}$$

У general и concatenation функцијама, W_a и V_a су тежинске матрице које се тренирају заједно са моделом. Користећи овако добијене скорове, Луонг је предложио два attention механизма; глобални и локални. Глобални приступ је једноставнији и он ће даље бити коришћен.

Глобални attention разматра све делове улазне секвенце, користећи информације из свих скривених стања енкодера. Ради бољег тумачења информација, користи се softmax функција за добијање пондерисаног просека по свим индексима вектора a_t и тако настаје вектор $a_t^{'}$, што је дато једначином (3.5).

$$a_t^{'} = \text{softmax}\left(a_t\right) \tag{3.5}$$

Коришћењем $a_t^{'}$ контекстни вектор се прерачунава једначином (3.6).

$$d_t = \sum_{t'=0}^{K} a_{t_{t'}}^{'} * h_{t'}^{E} \tag{3.6}$$

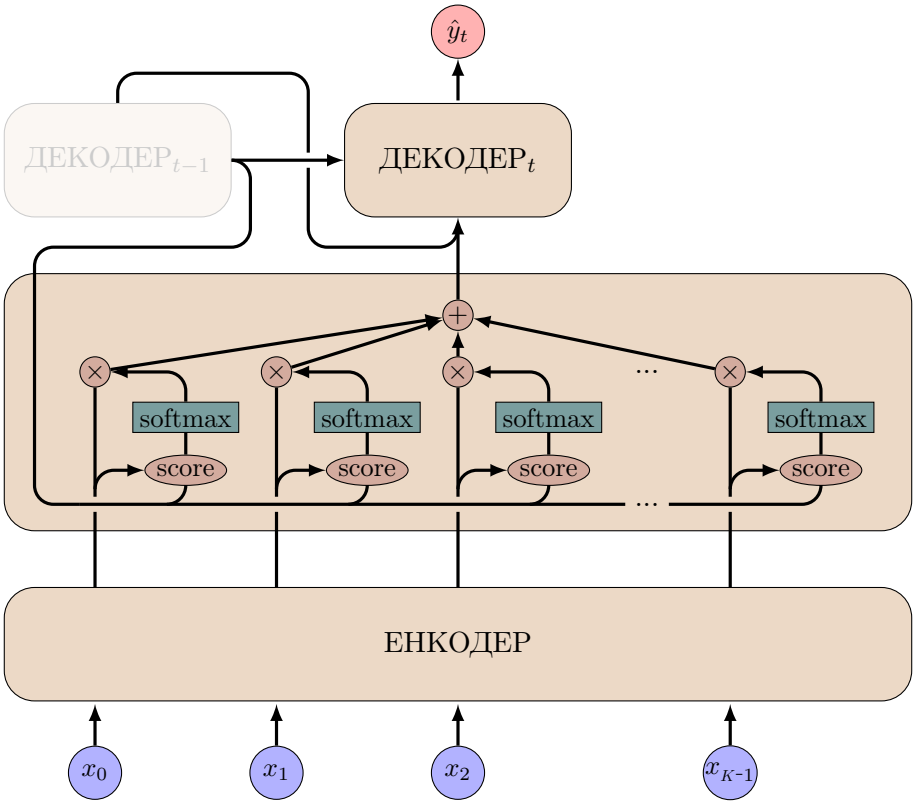
Контекстни вектор користи се заједно са скривеним стањем декодера за одређивање $\widetilde{h}_t^D$ вектора, који има улогу излазног скривеног стања. Израчунавање овог вектора дато је једначином (3.7).

$$\widetilde{h}_t^D = \tanh\left(W_{\widetilde{h}}\left[d_t, h_t^D\right]\right) \tag{3.7}$$

На крају, предвиђање се врши употребом softmax функције, што је дато једначином (3.8).

$$\hat{y}_t = \text{softmax}\left(W_y \widetilde{h}_t^D\right) \tag{3.8}$$

У следећем временском кораку $\widetilde{h}_t^D$ замењује h_t^D и процедура се понавља до завршног корака. Илустрација енкодер-декодер модела са attention механизмом дата је на слици 3.8.



Слика 3.8: Енкодер-декодер са attention механизмом

3.2.9 Teacher Forcing

Teacher Forcing је метода за брзо и ефикасно тренирање RNN модела неуронских мрежа који користе закључке из претходног временског корака као улаз. Ова метода критична је за развој модела дубоког учења језика који се користе у машинском превођењу, резимирању текста и наслављавању слика, међу многим другим апликацијама.

Приликом употребе декодера, у временском кораку t, највероватније предвиђање $\hat{y}_t$ одређује која реч постаје улаз у кораку $t + 1$. Међутим, такав приступ тренирање чини веома спорим и нестабилним.

Teacher forcing метода, у временском кораку t уместо да користи предвиђену реч из претходног корака $\hat{y}_{t-1}$, декодеру једноставно даје тачну реч y_{t-1}. Овај поступак значајно убрзава тренирање RNN-а уопште. Први пут је предложен као замена за BPTT 1989. године од стране Вилијамса *(енгл. Ronald J. Williams)* и Зипсера *(енгл. David Zipser)* [90].

У машинском превођењу које захтева пуно времена и скуп хардвер ова једноставна метода нарочито је корисна.

Глава 4

Припрема података

4.1 Корпусна лингвистика

Корпусна лингвистика је назив за метод у лингвистичким истраживањима у коме се користе велики узорци природног језика, било говорног или писаног, који се називају корпуси. Корпус је, најпростије речено, грађа. У ширем смислу, под овим термином се подразумева било каква збирка текстова сакупљена по одређеном критеријуму, а у ужем смислу та збирка текстова је у електронском формату, машински читљива, структурирана и самим тим циљано претражива.

4.1.1 Историјат корпусне лингвистике

Зачеци корпусне лингвистике била су пописивања вишеструких употреба речи и израза у текстовима. У почетку ови послови били су везани за пописивање речи из Библије и места у тексту где су се те речи појављивале. Прве конкордансе Библије датирају из XIII века и на њима је по правилу радио велики број монаха који су индексе речи правили ручно. Осим Библије, на исти начин су индексирана и друга дела. Пример представља Бекетов *(енгл. Andrew Becket) Concordance to Shakespeare* из 1787. године [60].

Буса *(итал. Roberto Busa)*, педесетих година XX века, почео је да ствара *Index Thomisticus*, индекс свих дела Томе Аквинског *(итал. Tommaso d'Aquino)*, који је касније пренесен на бушене картице и представља први корпус који се могао компјутерски обрађивати [54].

Веома важан био је рад лексикографа, који су речнике израђивали на основу примера стварне употребе језика. Џонсонов *(енгл. Samuel Johnson)* речник енглеског језика из 1755. године ослањао се на огроман корпус састављен од папирних трака са примерима употребе речи забележених између 1560. и 1660. године. *Оксфордов речник енглеског језика*, објављен 1884. године, израђен је на исти начин уз помоћ више од три милиона папирних трака [60].

Први електронски корпус писаног језика, *Браун корпус (енгл. Brown University Standard Corpus of Present-Day American English – Brown)*, настао је шездесетих година XX века. Записан је на бушеним картицама и садржао је око милион речи из текстова на енглеском језику са америчког говорног подручја. Материјал је прикупљен током 1961. године из 15 различитих језичких жанрова и био је намењен за потребе лингвистичке анализе [60].

Први корпус говорног енглеског језика израђен је на Универзитету у Единбургу између 1963. и 1965. године. Састојао се од 166.000 речи. Између 1975. и 1990. године израђен је *Лондон-Лунд корпус* говорног енглеског *(енгл. London-Lund Corpus of Spoken English – LLC)* који је садржао око пола милиона речи [60].

Седамдесетих година XX века постепено се увећавао број електронских корпуса. Поред енглеског, јављају се и корпуси на другим језицима, као и другачије врсте језичких корпуса. Током осамдесетих и деведесетих година корпусна лингвистика доживела је прави процват. Њен развој у овом периоду омогућили су све већа доступност компјутера и напредак технологије у погледу капацитета и брзине прикупљања и обраде података [60].

4.1.2 Класификација корпуса

Корпуси се могу сврстати у различите категорије на основу бројних параметара. Исти корпус може задовољавати критеријуме за више категорија.

Неке од подела могу бити [45]:

- Према намени:
  - **Општи**: Теже да представе читав језик, у свим његовим облицима, варијететима и стиловима. *(Brown[43], LOB[46], BNC[17])*
  - **Специјализовани**: Ограничавају се на одређени варијетет или само на одређене говорнике (дијалекатски, преводни, дечји, ученички и сл.). *(GPEC[8])*
- Према процедури селекције:
  - **Узорковани**: Садрже узорке текста сличне дужине, из различитих категорија, задржавајући балансираност и репрезентативност. *(Brown, LOB, SEU[70])*
  - **Комплетни**: Садрже читаве текстове. *(EPFTD[24])*
- Према динамици формирања:
  - **Статички**: Уколико се корпус након што се израда заврши више не мења. *(сви претходно наведени)*
  - **Динамички**: Уколико се корпус стално или повремено допуњује. *(BOE[6])*
  - **Колекције**: Нису у пуном смислу корпуси, због недостатка дизајна или намене, али ипак представљају велике скупове језичке грађе. *(Пројекат Гутемберг[26])*
- Према медијуму:
  - **Писани**: Садрже само писане текстове. *(Brown, LOB)*
  - **Говорни**: Састављени су од снимака, транскрипата и додатних ознака везаних за изговор, емоцију и слично. *(LLC[80])*
  - **Мултимодални**: Садрже и писани и говорни материјал. *(BOE, BNC)*
- Према броју језика/дијалеката:
  - **Једнојезични**: Уколико су сви укључени текстови написани на једном језику. *(сви претходно наведени)*
  - **Вишејезични** или **паралелни**: Садрже више језика/дијалеката. Паралелизам може бити различитог степена, од стриктно паралелног (оригинал и једна или више верзија превода, као што је *ENPC[32]*), до ниско паралелног (такозваних компарабилних корпуса), које чине колекције сличних текстова на више језика, као што је *ICE[30]*, добијен комбинацијом *Brown*, *LOB* и *Kolhapur[78]* корпуса.
- Према времену:
  - **Синхронијски**: Текстови су написани у једном временском периоду, обично у периоду прављења корпуса.
  - **Дијахронијски**: Садрже текстове из више временских периода. *(HC[71])*
- Према говорнику:
  - **Изворни**: Записи су продукт говорника којима је језик који се користи матерњи.
  - **Учени**: Записи су продукт говорника којима језик који се користи није матерњи.

- Према анотацији:
  - **Прост**: Корпус је настао скенирањем, без информација о самом тексту. Више се може сматрати колекцијом текста него правим корпусом. *(Пројекат Гутемберг)*
  - **Означен**:
    * **форматиран**: Корпус је подељен на странице, параграфе, са одређеним форматом, фонтом и сл. *(Brown)*
    * **са идентификацијом**: Садржи, аутора, жанр и сл. *(BNC, ICE-GB[83])*
    * **обележене врсте речи, дискурс и сл.** *(BNC, ICE-GB)*

4.1.3 Одабир корпуса за NMT

Подаци на којима ће модел бити трениран, њихов квалитет и бројност имају велики утицај на крајњи резултат. За потребе тренирања неуронске мреже која преводи текст са енглеског на српски језик потребан је корпус који се састоји од упарених текстова на енглеском и одговарајућих превода на српском језику. Осврнтом на класификацију може се закључити да би одговарајући био **стриктно парелелни корпус опште намене**. Постоји више начина за стварање оваквих корпуса, а за сваки је наведен пример доступних енглеско-српских корпуса, који имају бар 10^4 упарених превода.

Начини за формирање стриктно паралелних корпуса могу бити:

1. **Упаривање књига и њихових званичних превода** може се, генерално, према квалитету самог превода сматрати врло добрим извором. Међутим, неуронска мрежа, као и људски мозак, нема довољно капацитета да читав текст одједном научи и преведе. Текст је потребно поделити на краће целине и преводити га део по део. Због количине потребних података подела се мора аутоматизовати и ту се јављају први проблеми са формирањем паралелног корпуса на овај начин. Ако се подела књиге изврши по поглављима, сигурност да ће поглавље оригиналне књиге одговарати поглављу преведене не доводи се у питање, али поглавља су и даље предуга да би се преводила у целини. Следећи логичан потез био би подела по пасусима. Уз одбацивање оних који су дуги, пасус би био донекле адекватан. Решење ипак не може бити лако реализовано зато што сврха доброг превода тежи што природнијем укупном преносу смисла у односу на језик на који се преводи, више него ужем фокусу на пасус и реченицу. То узрокује да ужи контексти, као што су пасус и реченица, могу бити сасвим другачије преведени, или у другом редоследу написани у односу на оригиналне. Последица тога је чињеница да се ни пасус ни реченица не могу сматрати добро преведеним у ужем контексту, а проблем са упаривањем се такође знатно компликује и захтева надзор човека уз комплексан систем за аутоматизацију. Добар пример корпуса ове врсте је *Масивни паралелни корпус: Библија на 100 језика[16]*. Оно што се намеће као проблем при употреби овог корпуса је архаични језик који је коришћен. Још један релативно велики корпус ове врсте је *СрпЕнКор[42]* настао упаривањем више углавном књижевних дела домаће и светске литературе. Проблем са овим корпусом је његова недоступност за преузимање, а налог за претраживање је могуће добити, међутим претраге су лимитиране.

2. **Упаривање новинских чланака и њихових званичних превода** такође се може сматрати добрим извором података. Међутим, присутан је јако сличан приблем са паралелизацијом као и код књига, а стил који се обично користи у дневно-политичком извештавању често је јако формалан. Доступни корпуси ове врсте су: *SETIMES[52]* и *GlobalVoices [48]*.

3. **Упаривање титлова** написаних на различитим језицима за исти садржај спада у мање квалитетне податке, зато што су званичне верзије често тешко доступне, а аматерске неретко садрже пуно грешака. Велика предност овог приступа је стриктна паралелизација сама по себи, али само под условом да су титлови писани за исту верзију садржаја на који се односе. Количина података садржана у титлу просечног филма много је мања од количине података која би се добила из књиге, па је и овде аутоматизација у одабиру и упаривању неопходна. С обзиром на то да се на интернету налази огроман број различитих верзија једног садржаја за који се пишу титлови, из искуства се може рећи, а у пракси се показало, да аутоматизовано упаривање титлова често не даје стриктно паралелни корпус. Пример је *OpenSubtitles[48]* који је у потпуности аутоматизовано направљен и суочава се у великој мери са поменутим проблемима. Сличан пример представља *QED[1]*, корпус настао упаривањем титлова са *AMARA* едукативне платформе. Има знатно мање података, али такође и знатно мање проблема са паралелизацијом.

4. **Упаривање натписа** на производима написаним на различитим језицима спада у веома сигурне методе стварања стриктно паралелних корпуса. Најприступачнији подаци ове врсте су локализације софтверских производа отвореног кода. Мана овог приступа је често одсуство реченица у подацима. Међу адекватним корпусима ове врсте су: *KDE4, Ubuntu* и *GNOME [84]*.

5. **Упаривање записа симултаних превода** са званичних међународних састанака може дати јако добре резултате. Један од најпознатијих корпуса те врсте базиран је на дискусијама у *Европском парламенту [36]*. На жалост, не постоје слични корпуси који садрже српски језик.

6. **Писање корпуса** даје најбоље резултате, али уједно је и најзахтевније. Познати пројекат ове врсте који реализују волонтери зове се *Tatoeba[84]*, и садржи, између осталог, и скроман али јако квалитетан стриктно паралелни енглеско-српски корпус.

Сви наведени корпуси доступни за преузимање могу се наћи у пројекту *OPUS[84]*. Према наведеним карактеристикама и каснијим узорковањем линија скриптом из листинга 4.1 и њиховом анализом, утврђено је да одређени корпуси нису погодни за тренирање NMT модела. Најчешћи узроци су велика количина грешака у упаривању, архаични језик или одсуство реченица.

Листинг 4.1: randomSampler.sh

```
#!/bin/bash
cat $2 | perl -ne "print if (rand() < $1)"
```

Прихваћени корпуси наведени су у табели 4.1 и сортирани неопадајуће према броју српско-енглеских парова превода које садрже.

Корпус	Број докумената	Величина	Број парова
Tatoeba v20190709	1	0.86MB	13,513
GlobalVoices v2017q3	1,023	4.9MB	20,018
SETIMES v2	1	60.2MB	225,169
QED v2.0a	2,670	45.8MB	284,942
Σ	3,695	111.76MB	543,642

Табела 4.1: Одабрани корпуси

4.2 Пречишћавање података

По одабиру паралелних корпуса који задовољавају основне предуслове да би могли бити примењени у NMT-у, парове превода које они садрже треба адекватно уредити за ову намену. У истраживању Анастаскопулоса *(енгл. Antonios Anastasopoulos)* из 2019. године [3] приказан је утицај различитих врста грешака у тренинг корпусу на крајње перформансе NMT модела, на основу чега се намеће закључак да пречишћавање може направити јако велику разлику у крајњем резултату. У вези с тим, биће примењен оригинални приступ, који подразумева агресиван вид пречишћавања корпуса, који ће уместо покушаја исправке грешака које су неизбежне, чешће уклањати сумњиве преводе.

Преузети корпуси су форматирани у две датотеке, у једној су текстови на српском, у другој текстови на енглеском језику, где су упарени текстови они који се налазе на истим линијама. Ради боље прегледности, корисно је извршити спајање у једну датотеку, где ће парови бити у посебним редовима раздвојени табом, што се може учинити скриптом која је дата у листингу 4.2.

Листинг 4.2: concatenateCorpus.sh

```
#!/bin/bash
paste $1 $2 > $3
```

Узорковањем 0.001% садржаја *Tatoeba* корпуса спојеног у једну датотеку, добијени су парови приказани у листингу 4.3.

Листинг 4.3: Tatoeba 0.001% случајни узорак

```
Молим вас, пожурите. Please hurry.
Da li umeš da deaktiviraš bombu? Do you know how to deactivate a bomb?
Kupio sam neke namirnice. I bought some groceries.
Vi mora da ste novi ovde. You must be new here.
Хвала на исправци. Thanks for the correction.
Samo dajte sekund Tomu. Just give Tom a second.
Oni nisu mogli da putuju, jer se dogodio problem. They couldn't travel because a problem occurred.
Sâm sam naučio francuski. I taught myself French.
Učitelj je napisao nešto na tabli, ali nisam mogao da pročitam jer je bilo premalo. The teacher wrote something on the blackboard, but it was too small for me to read.
Ovde sam da bih igrao bejzbol. I'm here to play baseball.
Zaista sam srećan zbog toga. I'm really happy about it.
Треба да смањимо бирократију. We need to cut red tape.
Ne mogu sam da popravim auto. I'm not able to fix the car by myself.
```

Чак и на тако малом узорку могу се приметити разни потенцијални недостаци, као што су писање српског текста и ћирилицом и латиницом, постојање специјалних знакова и постојање скраћеница, који су у листингу 4.3 означени црвеном бојом. Даљим узорковањем уочава се све више проблема које треба отклонити да би се неуронска мрежа успешно тренирала.

4.2.1 Алфабет

Текстови од којих су састављени корпуси прикупљани су, углавном, у дугом временском периоду, различитим методама и из различитих извора. Наведене околности узрокују и појаву више различитих начина записивања у истом корпусу. Унификација кодирања слова српских алфабета може се постићи модификованом верзијом кодова које користи систем *АУРОРА* чији је аутор Витас [88]. Његова кодна схема пресликава дијакритичке карактере и диграфе који се користе у српској латиници на начин приказан табелама 4.2 и 4.3.

Велико	AURORA	ISO 8859-2	Мало	AURORA	ISO 8859-2	Пример
Č	CY, Cy	200	č	cy	232	Čačak = Cyacyak
Ć	CX, Cx	198	ć	cx	230	Ćićevac = Cxicxevac
Đ	DX, Dx	208	đ	dx	248	Đorđe = Dxordxe
Š	SX, Sx	169	š	sx	185	Šuškati = Sxusxkati
Ž	ZX, Zx	174	ž	zx	190	Žižak = Zxizxak

Табела 4.2: Дијакритички карактери

Велико	AURORA	ISO 8859-2	Мало	AURORA	ISO 8859-2	Пример
NJ, Nj	NX, Nx	—	nj	nx	—	Njegoš = Nxegosx
LJ, Lj	LX, Lx	—	lj	lx	—	Ljuljati = Lxulxati
DŽ, Dž	DY, Dy	—	dž	dy	—	Džordž = Dyordy

Табела 4.3: Диграфи

Предност оваквог начина кодирања је неутралисање разлика које потичу од различитих ћириличних и латиничних кодних система.

Недостаци оваквог кодирања долазе од:

1. недоследности у кодирању (нпр. *Džordž* се јавља и као *Dyordy* и као *Dzxordzx*);
2. графемских двосмислица (нпр. код римских бројева *CX* је и *110* и *Ć*).

Како проблеми са кодирањем иначе оптерећују домаће издаваштво, и како овакав начин кодирања своди текст на српском на 7-битни *ASCII* скуп карактера, даље ће бити коришћен *АУРОРА* систем кодирања.

Редослед слова у одабраном систему одређен је колационом секвенцом *ASCII* кода. То значи да ће, условно речено, реч *1%-тни* претходити речи *JEDNOPOSTOTNI* јер је код знака *1* мањи од кода знака *J*. Слично, реч *JEDNOPOSTOTNI* претходи речи *Jednopostotni* јер су им прва слова једнака, а *E* има мањи *ASCII* код од *e*.

Претварање ћириличних записа у *АУРОРА* енкодинг може бити решено претходним претварањем у латинични запис или директно. Специфично за задатак МТ-а корисно је читав текст записати малим словима, ради лашег разазнавања истих речи. Битно је напоменути да постоји више различитих енкодинга у којима се могу наћи слова специфична за српски језик. Неки од њих нису компатибилни са *UTF-8* енкодингом, у коме су записани преузети корпуси, и који ће их прочитати као: *è* и *æ*, уместо: *č* и *ć* респективно. Овакви проблеми нису чести у одабраним корпусима и парови који садрже ове и друге знакове који нису мала слова из *ASCII* енкодинга могу бити избачени, без великих губитака у материјалу.

Још једна врста симбола прави велике проблеме неуронским мрежама, а у питању су бројеви записани цифрама. Када би били третирани исто као и све друге речи, речник би се знатно увећао, без значајног добитка на садржајности, што није погодно. Имајући у виду да се бројеви у тексту могу писати и речима, што је у одређеном смислу и правилније, а има мањи утицај на проширивање речника, биће избачени сви парови који садрже арапске цифре.

Сами проблеми *АУРОРА* енкодинга могу бити у великој мери решени, узимајући у обзир претходно донесене одлуке. Први проблем може бити у потпуности решен тако што ће свака појава *Dž*, *DŽ* и *dž* бити записана као *dy*, па ће тек онда свако преостало *Ž* односно *ž* бити записно као *zx*. Други проблем у овом случају може бити делимично решен. Уз

констатацију да одабрани корпуси нису записани у *AУРОРА* енкодингу, могуће је одбацити све парове превода који у српском делу садрже било какве карактере који се не појављују у српској ћирилици и латиници и нису из скупа знакова за које је донесена одлука да буду задржани, а то су: ␣ , , , . , ? , ! , (,) , ’ .

Такође, корисно је одбацити и све парове који имају у потпуности исти запис и на српском и на енглеском језику. Узорковањем је примећено да код њих постоји одређена вероватноћа да су последица грешке у упаривању, а такви валидни парови у пракси су реткост.

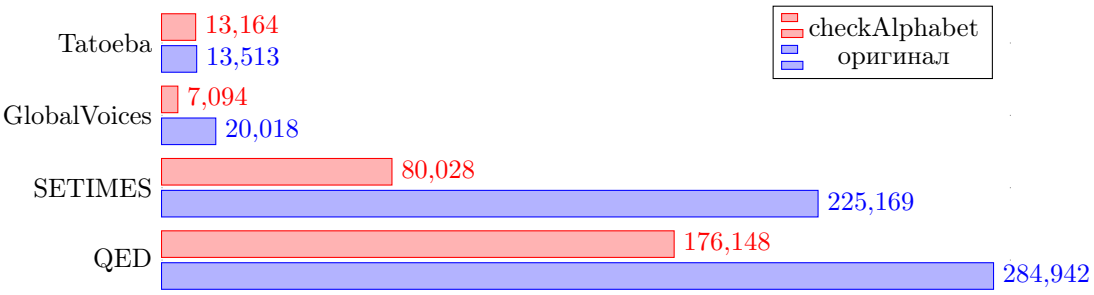
Овим приступом отклањају се и многи подаци из корпуса који могу бити проблематични за NMT јер садрже:

- арапске и римске цифре;
- управни говор;
- набрајања;
- стране речи;
- полусложенице;
- специјалне знакове;
- словне грешке;
- грешке у упаривању.

Провера којом се утврђује да ли пар превода садржи само дозвољене карактере може се извршити функцијом која је дата у листингу 4.4, а добијени резултати могу се видети на слици 4.1.

Листинг 4.4: checkAlphabet.py

```
def checkAlphabet(line):

  sr, en = line.split("\t")

  if sr == en:
    return False

  if re.search("[^"
    " .?!,’()"
    "АБВГДЂЕЖЗИЈКЛЉМНЊОПРСТЋУФХЦЧЏШабвгдђежзијклљмнњопрстћуфхцчџш"
    "ABCČĆDĐEFGHIJKLMNOPRSŠTUVZŽabcčćdđefghijklmnoprsštuvzž"
    "]", sr) is not None:

    return False

  if re.search("[^"
    " .?!,’()"
    "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
    "]", en) is not None:

    return False

  return True
```



Слика 4.1: Број превода који задовољавају 4.4

Објединьено превођење једне линије корпуса, односно једног пара превода, у *АУРОРА* запис може се постићи функцијом која је дата у листингу 4.5.

Листинг 4.5: modifyAlphabet.py

```
def modifyAlphabet(line):

    line = re.sub("А", "a", line)
    line = re.sub("Б", "b", line)
    line = re.sub("В", "v", line)
    . . .
    line = re.sub("Ч", "cy", line)
    line = re.sub("Џ", "dy", line)
    line = re.sub("Ш", "sx", line)

    line = re.sub("а", "a", line)
    . . .
    line = re.sub("ш", "sx", line)

    line = re.sub("Dž", "dy", line)
    line = re.sub("Ž", "zx", line)
    line = re.sub("Đ", "dx", line)
    line = re.sub("Lj", "lx", line)
    line = re.sub("Nj", "nx", line)
    line = re.sub("Ć", "cx", line)
    line = re.sub("Č", "cy", line)
    line = re.sub("Š", "sx", line)

    line = re.sub("dž", "dy", line)
    . . .
    line = re.sub("š", "sx", line)

    line = re.sub("A", "a", line)
    line = re.sub("B", "b", line)
    line = re.sub("C", "c", line)
    . . .
    line = re.sub("U", "u", line)
    line = re.sub("V", "v", line)
    line = re.sub("Z", "z", line)

    line = re.sub("X", "x", line)
    line = re.sub("Y", "y", line)
    line = re.sub("Q", "q", line)
    line = re.sub("W", "w", line)

    return line
```

4.2.2 Синтакса

При тренирању, NMT моделу треба обезбедити унос реченице, реч по реч у сваком временском кораку. Прву проверу коју треба извршити са тим у вези, јесте провера да ли је појединачни превод у форми једне реченице, и ако није треба га одбацити.

Да би било обезбеђено да свака реч буде једнозначно одређена треба извршити јасно одвајање речи једних од других и јасно одвајање интерпункцијских знакова од речи. То се може постићи увођењем правила да речи једне од других и речи од интерпункцијских знакова буду одвојене једним празним простором. Поред тога, неуронској мрежи било би тешко да научи где треба ставити зарез, те би било боље овај знак заменити празним простором. Даљим посматрањем долази се до закључка да много већи проблем представља употреба знака ’. У листингу 4.6 дат је 0.001% узорак *Tatoeba* корпуса, где су црвеном бојом означене речи које садрже овај знак.

Листинг 4.6: Tatoeba 0.001% случајни узорак

```
1 Odlazim. I’m going to go.
2 Prosto k’o pasulj. It’s a piece of cake.
3 Film počinje u deset sati. The movie starts at ten o’clock.
4 Документ је доспео у непријатељске руке. The document passed into the enemy’s hands.
5 Tomov dizajn je uštedeo kompaniji milione dolara. Tom’s design saved the company millions of
      dollars.
6 Mogao si da me poštediš dolaženja samo da si mi rekao da nisam ni trebao biti danas ovde. You
      could’ve saved me a trip if you’d just told me I didn’t need to be here today.
7 Oni nisu mogli da putuju, jer se dogodio problem. They couldn’t travel because a problem
      occurred.
```

У српском језику скраћенице које садрже ’ су ређе и такви преводи могли би се чак и одбацити. Међутим, у енглеском језику употреба овог знака је врло честа, а конструкције у којима се он појављује имају више облика, као што су скраћенице, ознаке на крају присвојних придева, негације глагола и други. Зато их је тешко на исти начин третирати. Због тога, без много компликација, биће направљен компромис и као у случају са зарезом, знак ’ ће бити замењен празним простором [11].

Уклањање заграда и садржаја унутар њих такође би било јако корисно, јер у заградама се налазе често синоними или објашњења одређених делова реченице, која нису неопходна, а могла би да представљају проблем при тренирању неуронске мреже.

На крају, више узастопних празних простора, узрокованих изменама насталим током пречишћавања, а и грешкама у самом корпусу, треба заменити једним празним простором. Такође, треба свакој линији у фајлу проверити структуру, која мора испуњавати захтеве да је сачињена од слова и размака, после којих следи интерпункцијски знак за крај реченице, затим таб, а затим још један скуп слова и размака, и на крају исти интерпункцијски знак као пре таба.

Овим поступком, поред уклањања зареза, апострофа, заграда и вишеструких празнина, отклањају се подаци из корпуса који садрже:

- преводе сачињене од више реченица;
- скраћенице са тачком;
- грешке у упаривању.

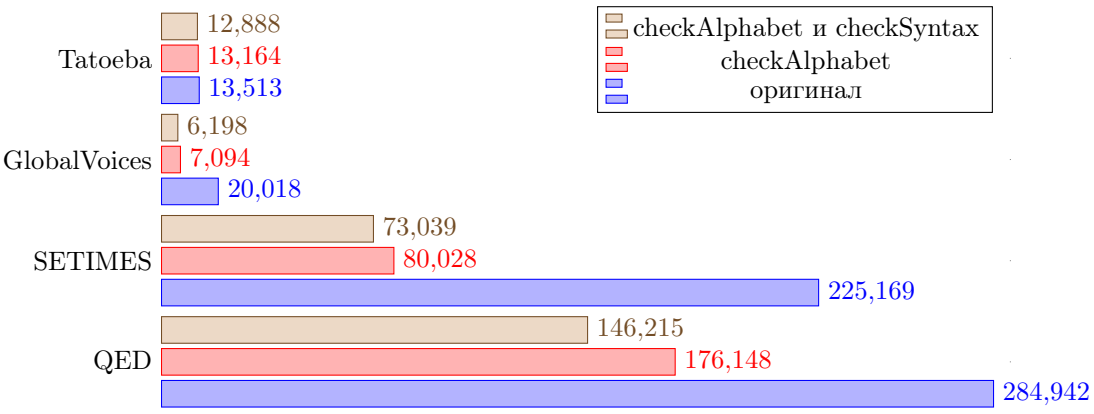
Функција која врши модификовање парова корпуса по описаној процедури дата је у листингу 4.7. Провера којом се утврђује да ли упарени преводи задовољавају установљена синтаксичка правила може се извршити функцијом која је дата у листингу 4.8, а резултати се могу видети на слици 4.2.

Листинг 4.7: modifySyntax.py

```
def modifySyntax(line):

    line = re.sub("\([^)]*\)", "", line)
    line = re.sub("[,']", " ", line)

    line = re.sub("\.", " .", line)
    line = re.sub("\?", " ?", line)
    line = re.sub("\!", " !", line)

    line = re.sub(" *\t *", "\t", line)
    line = re.sub(" +", " ", line)

    line = line.strip()

    return line
```

Листинг 4.8: checkSyntax.py

```
def checkSyntax(line):

    if re.search("(^[a-z ]+\.\t[a-z ]+\.$"
                 "|^[a-z ]+\?\t[a-z ]+\?$"
                 "|^[a-z ]+!\t[a-z ]+!$)", line) is None:
        return False

    return True
```



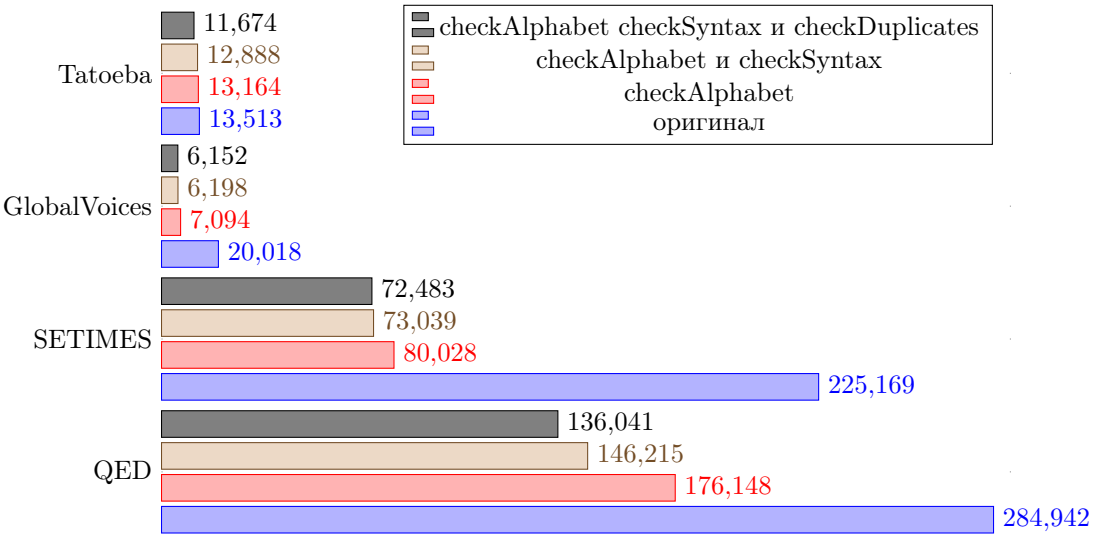
Слика 4.2: Број превода који задовољавају 4.4 и 4.8

4.2.3 Отклањање дупликата

Узорковањем је утврђено да се одређени парови превода више пута појављују, а с обзиром на релативно малу вероватноћу да два пара дужих превода буду апсолутно иста, може се доћи до закључка да је у питању пропуст у састављању корпуса. Ове врсте грешака могу утицати на погрешне закључке у вези фреквентности речи у корпусу, што ће у даљим корацима бити значајно. Дакле, било би добро приступити уклањању таквих дупликата, свесно притом уклањајући и дупликате кратких фраза које углавном имају идентичан превод на оба језика, а више пута се појављују. Функција која то ради дата је у листингу 4.9, а досадашњи резултати на слици 4.3.

Листинг 4.9: checkDuplicates.py

```
1 def checkDuplicates(lines):
2
3     return lines = list( set(lines) )
```



Слика 4.3: Број превода који задовољавају 4.4, 4.8 и 4.9

Већ после овог корака у пречишћавању може се видети да величина корпуса *GlobalVoices* постаје превише мала да би њиме био трениран NMT модел. Овај корпус даље неће бити разматран.

4.2.4 Дужина реченице

Са обновљеним интересовањем, дужина реченице се проучава од осамдесетих година XX века, углавном у контексту *других синтактичких феномена* [82]. Једна од дефиниција просечне дужине реченице прозног параграфа је однос броја речи и броја реченица [47].

У Корнејевом *(енгл. Andras Kornai)* уџбенику *Математичка лингвистика* сугерише се да је у новинарској прози просечна дужина реченице изнад 15 речи [39]. Просечна дужина реченице углавном служи као мерило за процену реченичне тешкоће или сложености [12]. Уопштено, ако се просечна дужина реченице повећа, онда ће се и сложеност реченице повећати [86].

Друга дефиниција дужине реченице је број клауза у реченици, док је дужина клаузе број гласова у клаузи [38]. Истраживања Шилса *(енгл. Erik Schils)* и де Хана *(енгл. Pieter de Haan)* базирана на узорцима текста су показала да две суседне реченице имају већу вероватноћу да имају сличне дужине од две реченице које нису суседне, и да готово сигурно имају сличну дужину у фикцији. Ово је супротстављено теорији да аутори могу тежити да имају наизменично дуге и кратке реченице [76]. Дужина реченице, као и сложеност речи, су фактори у читљивости реченице, међутим, и за друге факторе, као што је присуство везника, речено је да знатно олакшавају разумевање [2, 65].

Због свега наведеног има смисла ограничити дужину реченице у корпусу и на тај начин направити нове корпусе који садрже реченице лимитиране до одређене дужине, а самим тим и до одређене садржајне сложености. Биће коришћена једноставна метрика, и дужину реченице одређиваће број речи у њој. Из сваког од одабраних корпуса биће издвојена нова три корпуса које ће се састојати само од оних превода чија дужина реченице на оба језика не прелази 10, 20, односно 40 речи.

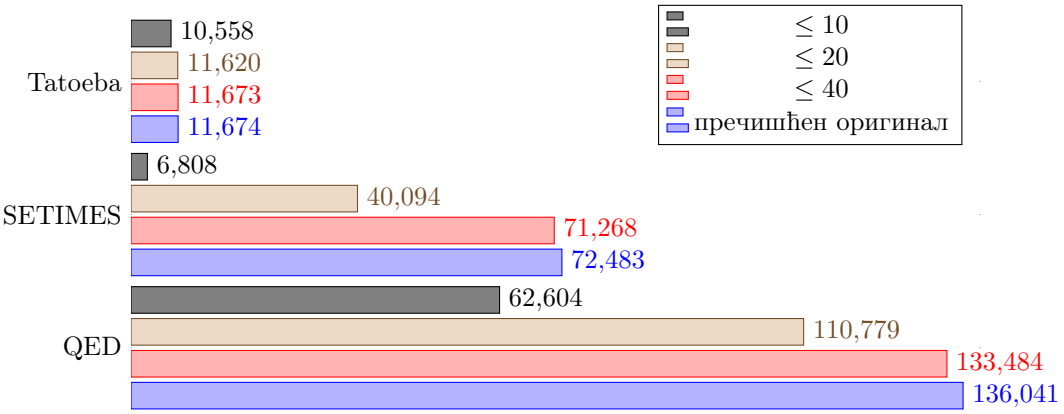
Овим поступком, поред саме поделе корпуса према претпостављеној комплексности, отклањају се и преводи који садрже:

- дуга набрајања;
- веома дуге реченице.

Провера дужине реченице може се извршити функцијом која је дата у листингу 4.10, а резултати се могу видети на слици 4.4.

Листинг 4.10: checkLength.py

```
1 def checkLength(line, n):
2
3     sr, en = line.split("\t")
4
5     if len( sr.split() ) > n or len( en.split() ) > n :
6         return False
7
8     return True
```



Слика 4.4: Број пречишћених превода који задовољавају 4.10 за дужине 10, 20 и 40

Након овог корака специфичност корпуса *SETIMES* долази до изражаја. Пошто су у питању преводи новинских чланака, где се углавном користе дуже реченице, подскуп овог корпуса у коме се појављују само реченице не дуже од 10 речи бива јако мали и даље неће бити разматран.

4.2.5 Лексика

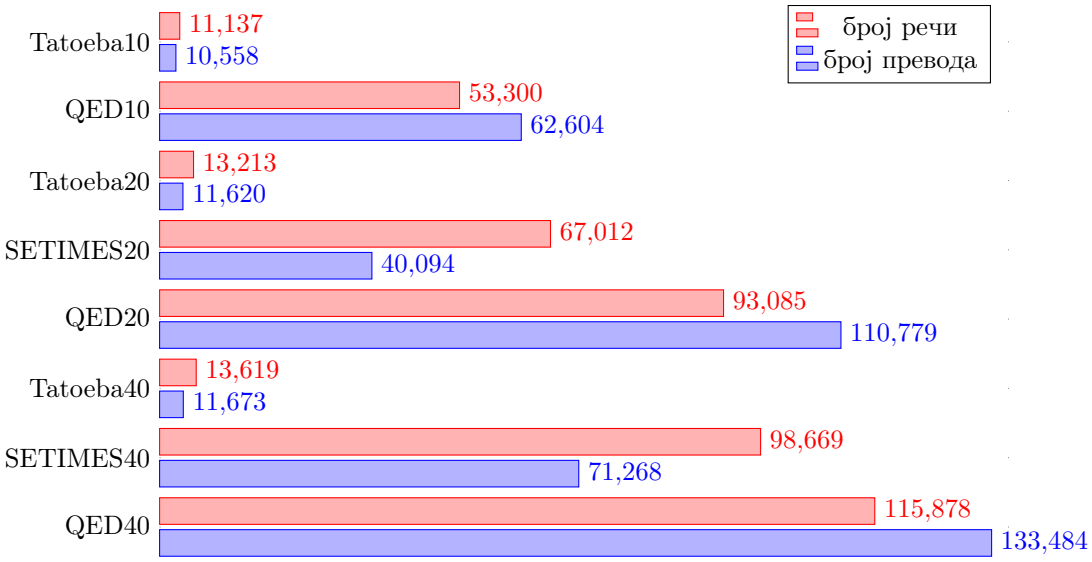
Број различитих речи у корпусу игра значајну улогу при тренирању неуронске мреже. Енкодер-декодер се може окарактерисати као врста класификатора која врши предвиђање наредне речи у сваком временском кораку. Што је број класа већи, у конкретном случају речник, то предвиђање постаје већи изазов.

Свака реч у NMT моделу памти се као вектор фиксне димензије, чији скалари бивају одређени на основу околних речи. Уколико корпус садржи велики број речи, то смањује могућности неуронске мреже да свакој речи одреди право место у датом простору, а самим тим и њено прецизно значење. Ситуација се компликује додатно ако корпус сам по себи није велик.

У листингу 4.11 дата је функција која одређује број различитих речи у корпусу и учесталост њиховог појављивања. На слици 4.5 приказан је однос броја речи и броја превода у до сада формираним и пречишћеним корпусима.

Листинг 4.11: countWords.py

```
def countWords(lines):

    counts = dict()
    words = " ".join(lines).split()

    for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1

    return counts
```



Слика 4.5: Број речи и превода у пречишћеним корпусима

У циљу превазилажења проблема узрокованих превеликим речником, одређен број речи потребно је одстранити. Најпогоднije би биле оне речи чије је значење најтеже одредити, а управо оне се најређе и појављују.

Ослањајући се на Зифов *(енгл. George K. Zipf)* закон [93] може се рећи да постоји велики број речи које се ретко појављују. Ова чињеница гарантује да се одстрањивањем најнефреквентнијих речи, а то су оне које се појављују само једном, значајно умањује речник корпуса и тако на више нивоа олакшава учење неуронској мрежи.

Приступ у коме би била вршена замена речи које се једном појављују специјалном речју не би проузроковао губљење података. Међутим, тај метод довео би до контаминације корпуса, јер би употребљена специјална реч постала превише честа, а обучена неуронска мрежа би јој током предвиђања давала висок приоритет. Агресивнији, али и ефикаснији начин је уклањање парова превода у којима се појављују поменуте речи. Додатно, овај поступак биће понављан све док у корпусу буду постојале речи које се једном појављују.

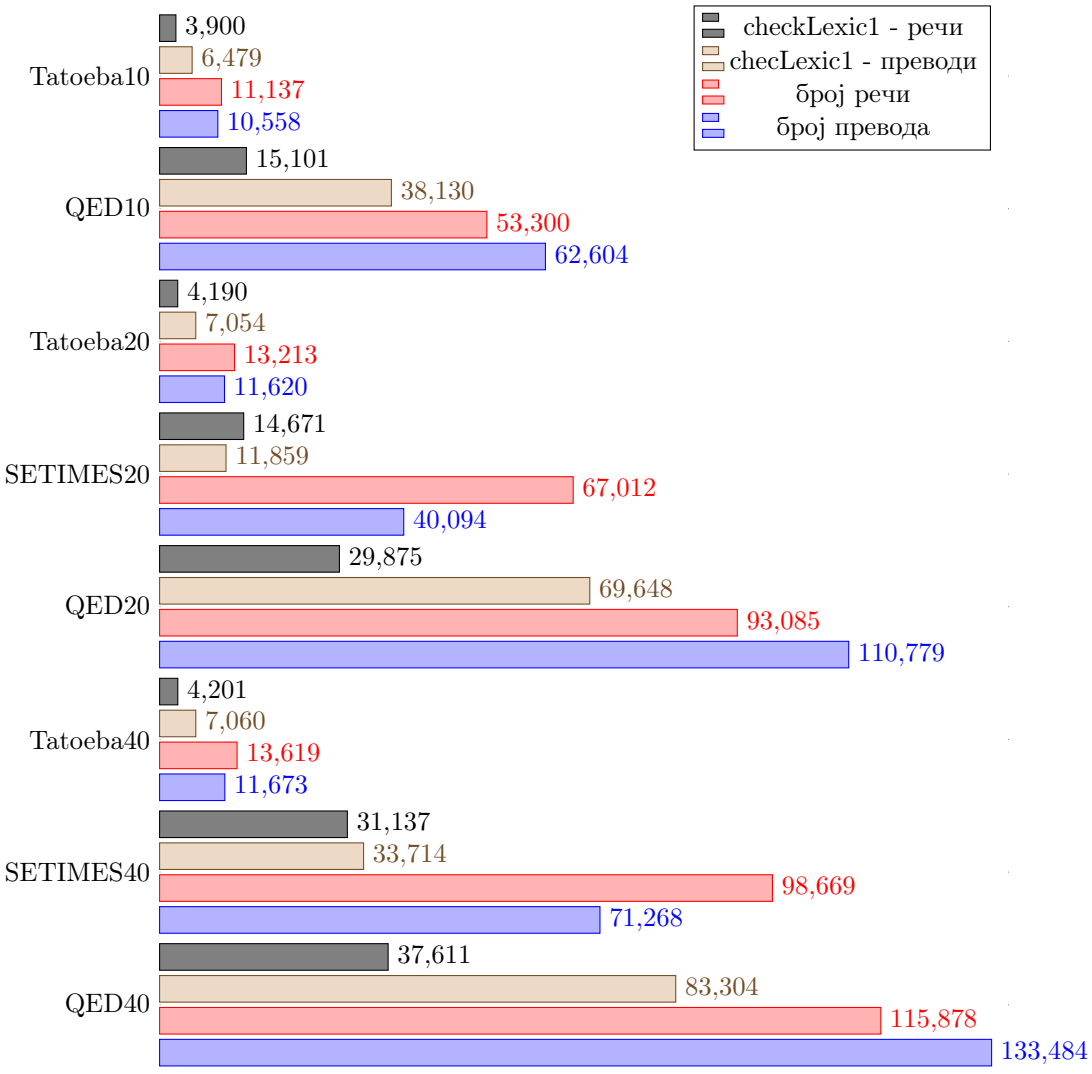
Овим поступком, осим умањивања речника, отклањају се и парови који садрже:

- словне грешке, које реч учине јединственом;
- властите именице које нису честе;
- облици речи који се ретко користе.

У листингу 4.12 дата је функција која врши пречишћавање на описан начин. На слици 4.6 приказан је новодобијени однос броја речи и парова превода у корпусима.

Листинг 4.12: checkLexic1.py

```
def checkLexic1(lines):

    clear = []; again = True

    while again:
        again = False; counts = countWords(lines)

        for line in lines:
            t = True
            for word in line.split():
                if counts[word] == 1: t = False; again = True; break
            if t: clear.append(line)

        lines = clear; clear = []

    return lines
```



Слика 4.6: Број речи и превода у пречишћеним корпусима који задовољавају 4.12

После ове фазе пречишћавања, може се приметити да је корпусима знатно више умањен број речи од броја превода, што је очекиван и добар резултат. Сви подскупови *Tatoeba* корпуса после овог корака остају са мање од 10^4 превода, али ће ипак бити задржани због поређења, а у недостатку алтернативе.

Посматрајући резултате са слике 4.6, може се видети да *QED* и *Tatoeba* корпуси после пречишћавања показују знатно повољнији однос броја речи и парова превода, мада *QED* у апсолутним бројевима и даље садржи много различитих речи. На примеру *SETIMES* корпуса, може се видети да код њега и даље постоји велики број речи у односу на број парова превода. Разлози за то могу се тражити у његовој специфичности, која укључује постојање великог броја властитих имена, а она, сама по себи, немају нарочито битно значење што се самог поступка превођења тиче.

Отклањање свих парова превода који садрже речи које се само два пута појављују у читавом корпусу постоји као опција. На тај начин било би уклоњено мање материјала него у претходном случају. Са друге стране сада већ има мање разлога за такав поступак, јер су грешке циљане овом процедуром већ уклоњене, а сама процедура би могла створити нове јединствене речи у корпусу.

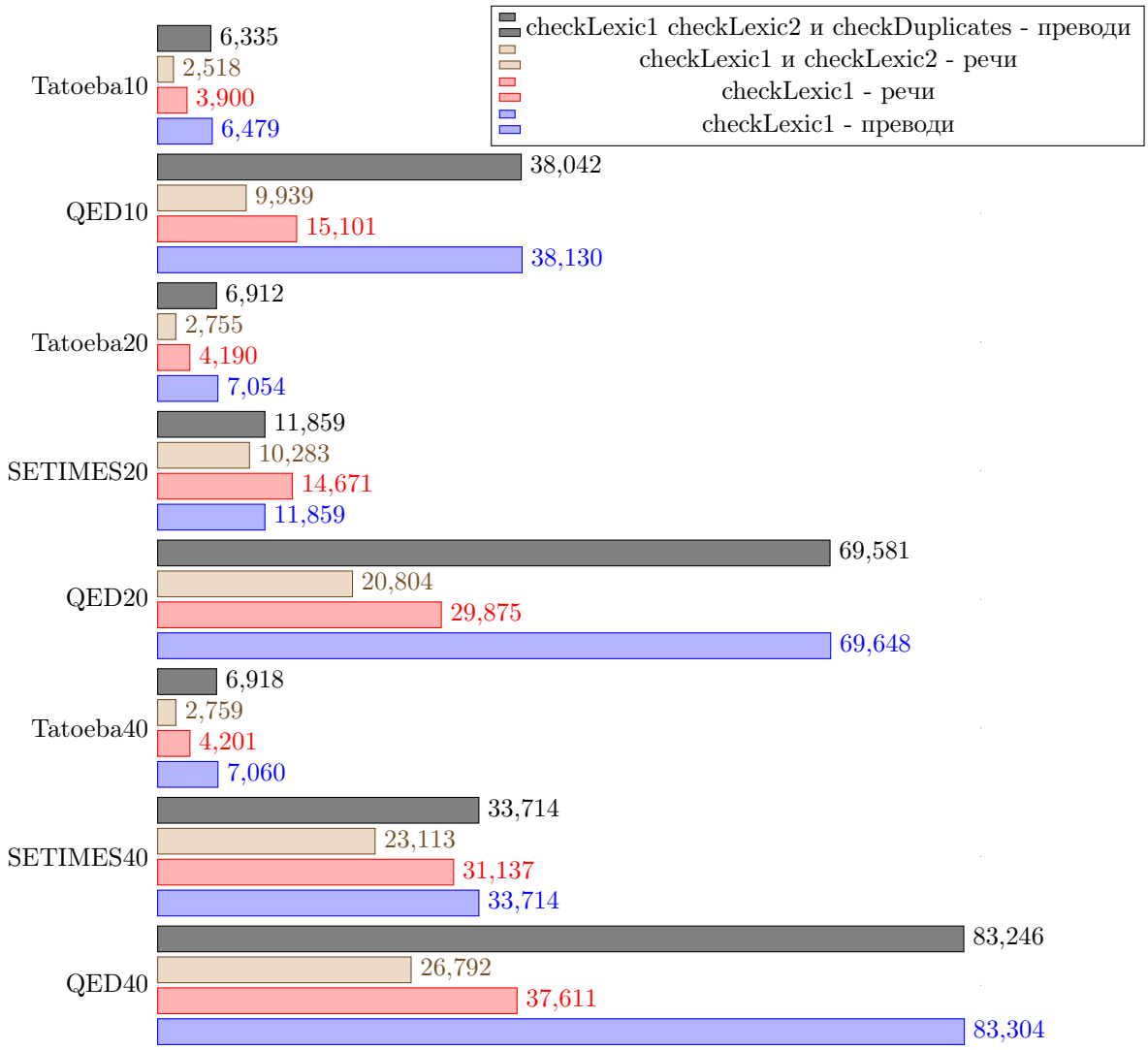
Уз претпоставку да ће се властито име појавити и у преводу и у оригиналу у истом облику (што не мора увек бити тачно), корекција корпуса може се извршити тако што ће свака реч која се појављује два пута бити замењена специјалним знаком $< UNK >$. Овај приступ, инспирисан властитим именицама, растеретиће корпус и од појаве неких других ретких речи. Што је можда и још значајније, на овај начин омогућава се неуронској мрежи да барата и са непознатим речима, тако што ће користити специјални симбол када се сусретне са речју која се у тренирању није ниједном појавила.

Напослетку, поново има смисла отклонити дупликате, функцијом датом у листингу 4.9, који овог пута могу бити узроковани појављивањем специјалног знака на одговарајућим местима у различитим преводима.

У листингу 4.13 дата је функција која врши описану модификацију. На слици 4.7 приказан је новодобијени однос броја речи и броја превода у пречишћеним корпусима.

Листинг 4.13: checkLexic2.py

```
def checkLexic2(lines):

    counts = countWords(lines)
    clear = []

    for line in lines:
        newLine = ""

        for word in line.split():
            if newLine != "":
                if newLine[-1] in ".?!":
                    newLine = newLine + "\t"
                else:
                    newLine = newLine + " "

            if counts[word] == 2:
                newLine = newLine + "<UNK>"
            else:
                newLine = newLine + word

        clear.append(newLine)

    return clear
```



Слика 4.7: Број речи и превода у пречишћеним корпусима који задовољавају 4.12 и 4.13

Поређењем података са слике 4.5 и слике 4.7 може се констатовати да су речници одабраних корпуса значајно умањени, а да је однос броја превода и речи доведен у погоднију сразмеру.

Овим кораком поступак пречишћавања би био завршен. Цена спроведених процедура је свакако осетан губитак података, али чињеница да су увек за уклањање бирани они подаци код којих постоји највећа вероватноћа да садрже грешку у доброј мери оправдава овакав приступ.

Скрипте које су коришћене за пречишћавање налазе се на јавном репозиторијуму [1].

[1] `https://gitlab.com/dragutinostojic/nmtsren/-/tree/master/clearCorpus`

Глава 5

Експеримент

Обрада природних језика *(енгл. Natural Language Processing – NLP)*, до сада, није у великој мери примењивана на јужнословенске језике. Специфичности ових језика, као и недостатак истаживања, представљају додатни изазов у многим задацима из ове области.

У случају машинског превођења са енглеског на српски, циљни језик је сложенији од изворног у неколико аспеката. Српски језик, као и други словенски језици, морфолошки је богат и има прилично слободан редослед речи. Даље, за разлику од других словенских језика, српски има два писма, па треба обратити пажњу како се писма не би мешала у једном корпусу. Друга могућа недоследност у корпусима је различито руковање властитим именицама - на ћирилици је могућа само транскрипција, док су на латиници присутни и транскрипција и писање оригинала. Поред тога, све именице мењају се по падежима, род им се не може претпоставити и слично.

Један од првих значајних радова на нашим просторима из области MT-а је *Presis* RBMT систем из 2002. године, који су увели Ромих и Холозан [73]. Први радови везани за SMT су енглеско-српски систем за превођење које је увела Поповић и остали 2005. године [69] и словеначко-енглески које је увела Маучец и остали 2006. године [53]. Прве резултате у превођењу са хрватског на енглески објавио је Љубешић и остали 2010. године, на малом скупу везаном за временску прогнозу [51], док је 2014. Торал са другима објавио машинског преводиоца који се фокусирао на домен туризма [85]. SMT системи за превођење титлова који су укључивали српски и словеначки језик појавили су се као део *SUMAT* пројекта 2014. године [20].

Сва претходно поменута истраживања рађена су на малим корпусима. Први покушај истраживања са већим количинама података вршили су Поповић и Арчан 2015. године, а прикупљани су паралелни корпуси српског и словеначког језика [68].

Сви претходно поменути системи нашли су примену само у јако уским доменима. Арчан, Поповић и Битерал *(хол. Paul Buitelaar)* 2016. године објавили су SMT систем *Asistent* који је у то време дао значајније резултате у превођењу између енглеског, српског, хрватског и словеначког језика без одређеног домена [4].

Поповић је 2018. године вршила испитивања односа SMT-а и NMT-а у проблему који је укључивао енглеско-немачки и енглеско-српски превод [67]. Лохар *(енгл. Pintu Lohar)*, Поповић и Вај *(енгл. Andy Way)* су 2019. године објавили резултате истраживања које је укључивало прављење енглеско-српског машинског преводиоца за корисничке рецензије филмова са сајта *IMDB* коришћењем NMT приступа [49].

Циљ овог мастер рада је прављење NMT модела, који преводи са енглеског на српски језик. По угледу на досадашње резултате и најбоље примере из праксе, тежиће се моделу солидних перфомансi уз хардверске захтеве за тренирање које може испунити и кућни рачунар. Резултати ће бити поређени са сложеним и хардверски изузетно захтевним NMT системима иза којих стоје велике корпорације, а који су доступни преко *Google Translate*, *Microsoft Translator* и *Yandex Translate* сервиса.

5.1 Модел

5.1.1 Окружење

Имплементација алгоритама за тренирање неуронских мрежа и њихово моделовање може били врло захтеван задатак. Временом је развијено неколико популарних библиотека за дубинско учење које умногоме олакшавају овај процес.

Најпознатије библиотеке за машинско учење су:

- *Theano* - развијен на Универзитету у Монтреалу 2007. године;
- *TensorFlow* - развијен од стране компаније *Google* 2015. године;
- *PyTorch* - развијен од стране компаније *Facebook* 2016. године.

На њиховој основи даље су настајале разне библиотеке у којима су уведени виши степени апстракције. Једна од популарнијих библиотека ове врсте је *Keras*.
За потребе NLP-а и NMT-а, данас постоји више специјализованих програмских оквира *(енгл. Framework)*.

Неки од најпознатијих програмских оквира за NMT су:

- *seq2seq* - енкодер-декодер оквир генералне намене настао на идеји MT-а, а базиран на *TensorFlow* библиотеци;
- O*penNMT* - екосистем специјализован за NMT и учење секвенци, који може радити на *TensorFlow* и *PyTorch* библиотеци;
- *Neural Monkey* - алат специјализован за NLP, базиран на *TensorFlow* библиотеци;
- *NEMATUS* - енкодер-декодер модел са attention механизмом базиран на *TensorFlow* библиотеци.

Иако велике компаније често стоје иза њих, сва поменута софтверска решења су потпуно бесплатна и отвореног су кода, што доприноси бржем развоју њих самих и екосистема који их окружују.
Популаран језик за програмирање у овој области је *Python*, чему доприноси његова једноставност, подршка и огроман екосистем, што све заједно омогућава више фокусирања на проблем него на имплементацију. Менаџер за управљање пакетима, као што је *Conda*, омогућава инсталацију *Python* окружења у дељеним вишекорисничким окружењима, као што су универзитетски кластери, на којима корисници имају ограничене привилегије. Због свега наведеног овај језик је изабран у тада актуелној верзији 3.8.
Што се тиче програмског оквира, иако би неко од специјализованих решења знатно убрзало процес развоја, донета је одлука да се он не користи. Разлози за то су пре свега ограничења у слободи имплементације и мања контрола позадинских активности које би могле да ставе добијене резултате под знак питања и да угрозе перформансе.
Уместо тога биће коришћена библиотека нижег нивоа. Међу три наведене, избор је пао на *PyTorch* који је преузет у тада актуелној верзији 1.5. Разлози су одлична интеграција са *Python*-ом и чврста спрегнутост са управљачким софтвером графичких процесора, која резултује јако добрим перформансама код тренирања.
Што се хардвера тиче, коришћен је *NVIDIA TITAN Xp GPU* са 12GB меморије, драјвером верзије 440.82 и *CUDA* драјвером 10.2, *Intel Core i5-6400 CPU @ 2.70GHz* и 8GB RAM меморије.

5.1.2 Архитектура

Архитектура самог модела направљена је на основу актуелних истраживања у области NMT-а, где је притом вођено рачуна о ниским хардверским захтевима.

Како би се што боље применили концепти о којима је било речи у овом раду, почетне тачке у имплементацији биће Робертсонов *(енгл. Sean Robertson)* пројекат *Translation with a Sequence to Sequence Network and Attention*, који је постављен као званично упутство PyTorch тима за прављење NMT модела 2017. године [72] и његова унапређена верзија коју је у својој тези представио Ланерс *(енгл. Quinn M. Lanners)* 2019. године [44]. Такође, као битна референца, коришћен је и рад *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,* у коме је компанија *Google* представила свој NMT систем 2016. године [91].

Резултат овог рада је једноставна и ефектна имплементација енкодер-декодер модела, која користи глобални attention механизам и двосмерни LSTM. Поред тога употребљен је teacher forcing стил обучавања и омогућен избор између CPU и GPU процесора, такође и опција за груписање података како би се убрзао процес обуке.

У раду објављеном 2017. године који носи назив *Massive Exploration of Neural Machine Translation Architectures*, *Google Brain* тим, са великим ресурсима које поседује компанија, тестирао је много различитих NMT модела и изнео добијене закључке [9]. Врло је значајна констатација да више, у задатку MT-а, не мора нужно значити и боље. Пример за то је откриће да повећавање дубине неуронске мреже, после одређене тачке, може довести не само до повећавања времена потребног за тренирање, већ и до лошијег резултата. На основу запажања, како из овог, тако и из других поменутих истраживања, као и многобројних експеримената везаних за конкретан случај, заокружен је дизајн и одређене су вредности хиперпараметара експерименталног модела, што је представљено у табели 5.1.

Хиперпараметар	Експериментални модел
RNN тип	LSTM
Величина embedding вектора	512
Величина контекстног вектора	512
Енкодер	двосмерни
Дубина енкодера	2
Дубина декодера	2
Attention механизам	глобални
Критеријум губитка	NLLLoss
Величина batch-а	128
Dropout	0.2
Оптимизатор	Adam
Learning rate	0.0002
Критеријум заустављања	16 епоха стагнације BLEU скора

Табела 5.1: Параметри модела

Препорука да се користи од 2 до 4 LSTM слоја [9] у енкодеру и декодеру тестирана је на припремљеним подацима и није забележено побољшање резултата на дубинама већим од 2. Разлог може бити релативно мала величина коришћених корпуса.

Оптимизатор *Adam* са брзином учења 0.0002 [91], показао је боље резултате и брже конвергирање од истог оптимизатора са брзином 0.0001 [9]. Разлог се, вероватно, налази у већој шанси за излаз из локалног минимума при већем коефицијенту брзине учења.

Величина batch-а у поменутим истраживањима креће се у распону од 10 [44] до 128 [91, 9]. Тестирањем је утврђено да нешто мању стабилност, али убедљиво најкраће време тренирања даје величина 128.

Подаци који су коришћени описани су у поглављу 4. Током обуке и тестирања, преводи

у сваком корпусу уређени су на случајан начин и подељени у размери 1 : 9 где се већи део користи за обучавање, а мањи за тестирање. Пре почетка сваке епохе, зарад стабилности модела, мења се редослед преводима у оба скупа, на случајан начин. Услед релативно мале количине података, нема простора за валидациони скуп и тренирање је вршено без њега, што је пракса у таквим ситуацијама. Због тога заустављање се врши када се догоди да нема увећања BLEU скора 16 епоха у односу на постигут максимум, а стање модела у коме се догодио најбољи разултат сматра се коначним.

Сваки тест скуп, на случајан начин издвојен из корпуса, на коме је тестиран модел, преведен је и помоћу *Google Translate*, *Microsoft Translator* и *Yandex Translate* сервиса. Само превођење било је проблематично због релативно велике количине текста и установљених лимита које прописују компаније. Међутим, коришћењем алтернативних приступа, уместо директног обраћања веб сервисима, као што је *Microsoft Office* пакет у случају *Microsoft Translator*-а, или сервис за превођење веб страна који нуди *Yandex*, могуће је те лимите значајно повећати, што у комбинацији са довољно стрпљења подухват чини изводљивим. Добијени преводи форматирани су на идентичан начин као и превод експерименталног модела, а затим им је одређен BLEU скор и извршено поређење. За одређивање BLEU скора коришћена је функција *corpus_bleu* из *nltk.translate.bleu_score* библиотеке. Сви резултати приказани у наставку добијени су на основу три понављања процеса обуке, а представљен је други најбољи резултат.

Постоји још много техника које се користе да би био побољшан квалитет превода. Пример су: beam search који може унапредити резултате претрагом стабла предвиђања и одабиром највероватнијег, употреба споријег али прецизнијег оптимизатора, постепена деградација хиперпараметра брзине учења *(енгл. decay)*, комбинација оптимизатора и многе друге. Ове и сличне методе ипак нису употребљене, да би се задржала једноставност и омогућио ефикасан процес обуке, а добијени модел, као чиста основа, оставља могућност даљег унапређивања различитим техникама.

Имплементација модела и пратећих процедура налази се на јавном репозиторијуму [1].

5.2 Метрика

5.2.1 NLLLoss

NLLLoss функција губитка *(енгл. Negative Log Likelihood Loss – NLLLoss)* погодна је за тренирање класификационих модела који треба да разликују већи број класа.

У пракси излазни softmax слој користи се у спрези са $NLLLoss$ функцијом. Испоставља се да је ова комбинација јако добра, што због резултата класификације, што због лакоће диференцирања. Израчунавање функције губитка на овај начин дато је једначином (5.1).

$$L_{NLLLoss} = -\frac{1}{n}\sum_{i=0}^{n-1} y_i^\top \ln \hat{y}_i = -\frac{1}{n}\sum_{i=0}^{n-1}\sum_{k=0}^{K-1} y_{i_k} \ln \hat{y}_{i_k} \tag{5.1}$$

Током процеса обуке умањује се вредност функције губитка поправљањем параметара модела, који представљају тежине веза. Губитак се може тумачити као мера незадовољства у односу на тренутне параметре. Што је већи губитак, то је веће и незадовољство.

Код употребе $NLLLoss$ као функције губитка, која дате вредности ставља у опсег $(0, +\infty)$, сумирајући шансе за све исправне класе, оно што се заправо догађа је да кад год мрежа додељује високу поузданост исправној класи, незадовољство параметрима је мало, али када мрежа додељује ниску поузданост исправној класи, незадовољство постаје велико.

Ова функција представљаће интерни скор на основу кога ће модел бити трениран.

[1] `https://gitlab.com/dragutinostojic/nmtsren/-/tree/master/`

5.2.2 Перплекситет

У теорији информација, перплекситет *(енгл. Perplexity)* је мера која одређује колико добро дистрибуција или модел вероватноће предвиђа узорак. Користити се и за поређење различитих модела. Низак перплекситет указује да је способност предвиђања добра.

Перплекситет је експоненцијална функција ентропије модела H и у конкретном случају уско је повезан са $NLLLoss$ функцијом губитка. Перплекситет се, с обзиром на то, може рачунати једначином (5.2).

$$Perplexity = e^{H} = e^{NLLLoss} \tag{5.2}$$

Због ове повезаности, перплекситет не даје много додатних информација. Међутим, промене у перплекситету оштрије се одражавају на графицима и лакше су за праћење.

5.2.3 BLEU

BLEU *(енгл. Bilingual Evaluation Understudy)* је алгоритам којим се мери разлика између аутоматског превода и једног или више референтних превода исте изворне реченице које је створио човек [62].

BLEU алгоритам упоређује n-граме аутоматског превода са n-грамима референтног превода, а затим броји подударања дајући им на значају у односу на њихову дужину. Подударања су независна од позиције. Виши степен подударања указује на већи степен сличности са референтним преводом и даје већи резултат. Разумљивост и граматичка исправност текста се не узимају у обзир.

Употребна вредност BLEU скора огледа се у јакој корелацији са људском проценом квалитета превода у појединачним реченицама на тестном корпусу, притом не укључујући компликоване прорачуне нити разумевање тих реченица. Рачунање BLEU скора врши се једначином (5.3).

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \ln p_n\right) \tag{5.3}$$

p_n представља модификовану прецизност n-грама која се добија као количник укупног броја појављивања датог n-грама у оригиналном и добијеном преводу, $w_n \in [0, 1]$ је важност дата одређеном n-граму, док је N број n-грама који улазе у прорачун. Уобичајно важи $N = 4$ и $w_1 = w_2 = w_3 = w_4 = 0.25$.

BLEU скор има тенденцију да фаворизује краће преводе. Да би се решио тај проблем уведени су пенали за кратке преводе *(енгл. Brevity Penalty – BP)* и описани једначином (5.4).

$$BP = \begin{cases} 1, & c > r \\ \exp\left(1 - \frac{r}{c}\right), & c \leq r \end{cases} \tag{5.4}$$

c представља број речи у реченици кандидату, а r је најбоље поклопљена дужина сваког кандидата у корпусу.

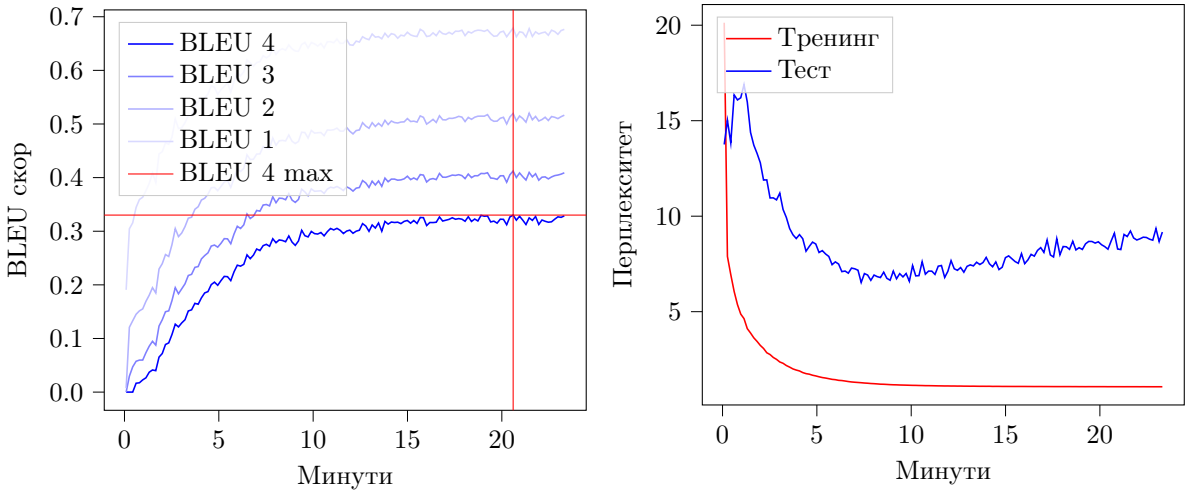
BLEU скор јако зависи од ширине домена, усклађености тестних података са тренинг подацима и количине података која је на располагању. Ако су модели обучени на уском домену, а тренинг подаци у складу са тестним подацима, може се очекивати висок резултат.

Постоји још много метрика које се данас користе, али BLEU је због своје једноставности, ефикасности и могућности поређења са другим резултатима одабран као најзначајније мерило перформанси у овом експерименту.

5.3 Резултати

5.3.1 Tatoeba 40

Наменски писан корпус *Tatoeba*, пречишћен и ограничен на реченице не дуже од 40 речи. Садржи 2,759 речи и 6,918 парова превода.



(а) BLEU скор током обуке

(б) Перплекситет током обуке

Слика 5.1: Понашање модела током обуке

	BLEU 1 (%)	BLEU 2 (%)	BLEU 3 (%)	BLEU 4 (%)
Google Translate	69.16	53.31	42.97	**35.26**
Microsoft Translator	65.99	49.76	39.77	32.65
Yandex Translate	56.17	37.96	27.58	20.58
Експериментални модел	67.88	51.97	41.23	33.01

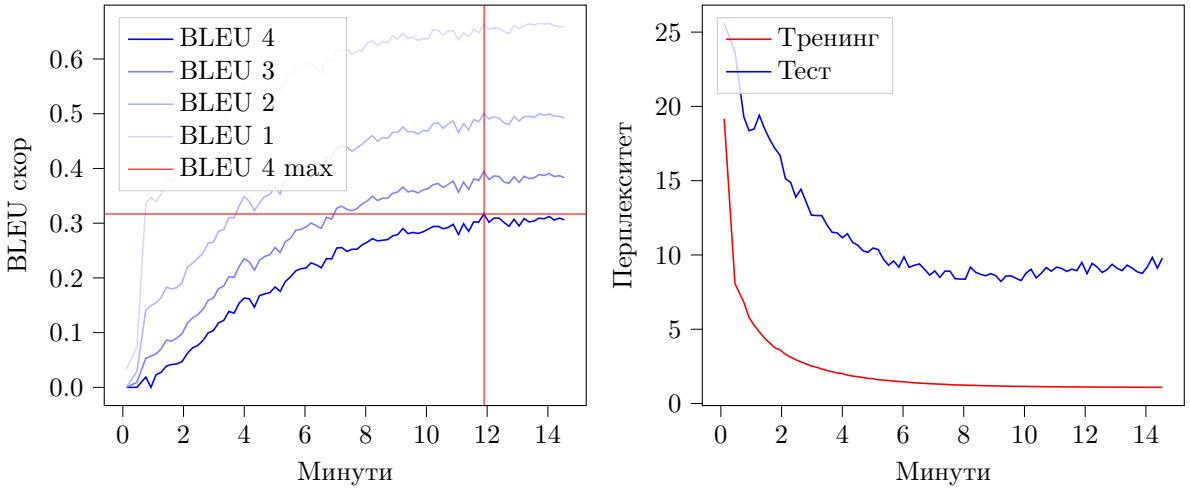
Табела 5.2: Поређење резултата за *Tatoeba 40* тест корпус

Листинг 5.1: Узорак *Tatoeba 40* тест корпуса преведен експерименталним моделом

```
> i was just heading out .
= taman sam krenuo .
< samo sxto sam krenuo . <EOS>

> i was happy yesterday .
= jucye sam bio srecxan .
< jucye sam bila srecxna . <EOS>

> tell me why you did it .
= kazxite mi zasxto ste uradili to .
< reci mi zasxto si uradila to . <EOS>

> i m sure that i ll win that tennis match .
= siguran sam da cxu dobiti taj teniski mecy .
< sigurna sam da cxu da <UNK> na tom teniskom mecyu . <EOS>
```

5.3.2 Tatoeba 20

Наменски писан корпус *Tatoeba*, пречишћен и ограничен на реченице не дуже од 20 речи. Садржи 2,755 речи и 6,912 парова превода.



(а) BLEU скор током обуке

(б) Перплекситет током обуке

Слика 5.2: Понашање модела током обуке

	BLEU 1 (%)	BLEU 2 (%)	BLEU 3 (%)	BLEU 4 (%)
Google Translate	69.67	55.37	43.23	**35.81**
Microsoft Translator	65.56	49.50	39.61	32.58
Yandex Translate	54.45	36.06	26.21	19.99
Експериментални модел	66.30	50.02	39.39	31.68

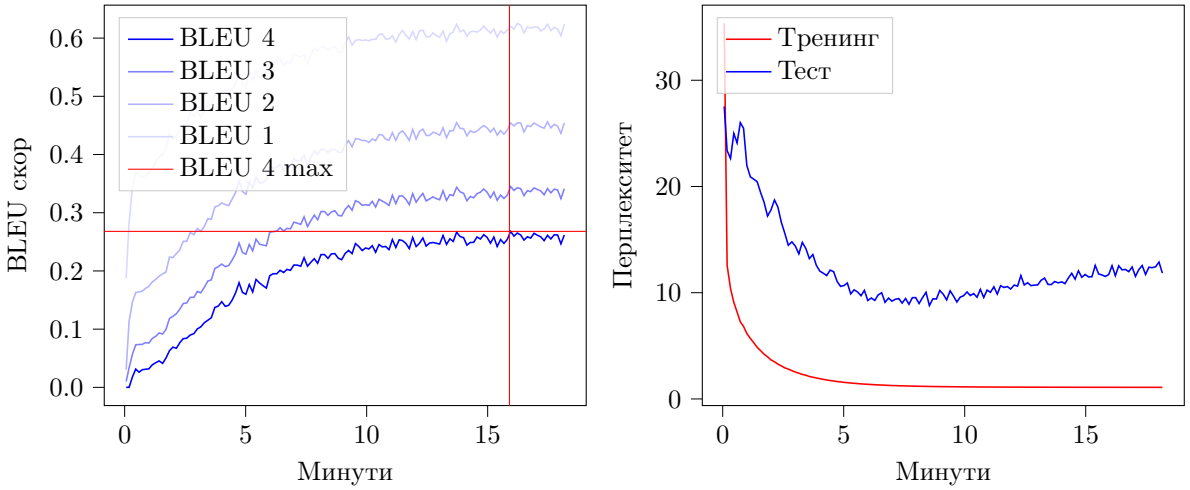
Табела 5.3: Поређење резултата за *Tatoeba 20* тест корпус

Листинг 5.2: Узорак *Tatoeba 20* тест корпуса преведен експерименталним моделом

```
> i m in better shape than you are .
= u bolxoj sam formi nego ti .
< u bolxoj sam formi nego vi . <EOS>

> i ve just been too busy .
= samo sam bio <UNK> .
< samo sam bio previsxe zauzet . <EOS>

> you are not allowed to go into that room .
= nemasx dozvolu da udxesx u tu prostoriju .
< nemate dozvolu da udxete u tu sobu . <EOS>

> get lost !
= gubi se !
< mrdaj se ! <EOS>
```

5.3.3 Tatoeba 10

Наменски писан корпус *Tatoeba*, пречишћен и ограничен на реченице не дуже од 10 речи. Садржи 2,518 речи и 6,335 парова превода.



(а) BLEU скор током обуке

(б) Перплекситет током обуке

Слика 5.3: Понашање модела током обуке

	BLEU 1 (%)	BLEU 2 (%)	BLEU 3 (%)	BLEU 4 (%)
Google Translate	68.87	52.08	41.26	**33.57**
Microsoft Translator	64.79	47.60	37.27	29.90
Yandex Translate	54.86	35.99	25.09	17.68
Експериментални модел	62.04	45.42	34.50	26.80

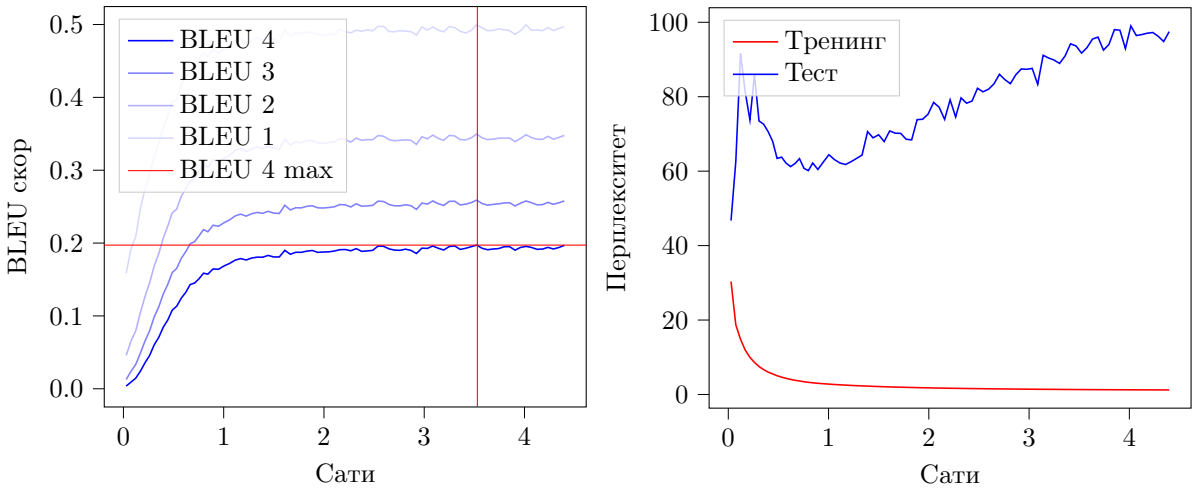
Табела 5.4: Поређење резултата за *Tatoeba 10* тест корпус

Листинг 5.3: Узорак *Tatoeba 10* тест корпуса преведен експерименталним моделом

```
> come and sit down .
= dodxi i sedi .
< dodxi da <UNK> i mene . <EOS>

> we better tell the others .
= bolxe da kazxemo <UNK> .
< treba da razgovaramo o ime . <EOS>

> am i ever wrong ?
= gresxim li ja ikad ?
< da li ja ikada <UNK> da sam ? <EOS>

> i don t like it one bit .
= nimalo mi se ne dopada .
< nimalo mi se ne svidxa . <EOS>
```

5.3.4 SETIMES 40

Корпус *SETIMES*, сачињен од превода реченица из новинских чланака, пречишћен и ограничен на реченице не дуже од 40 речи. Садржи 23,113 речи и 33,714 парова превода.



(а) BLEU скор током обуке

(б) Перплекситет током обуке

Слика 5.4: Понашање модела током обуке

	BLEU 1 (%)	BLEU 2 (%)	BLEU 3 (%)	BLEU 4 (%)
Google Translate	62.39	49.27	40.16	33.12
Microsoft Translator	74.08	64.79	57.90	**52.27**
Yandex Translate	56.66	41.73	31.87	24.62
Експериментални модел	49.97	34.97	25.89	19.72

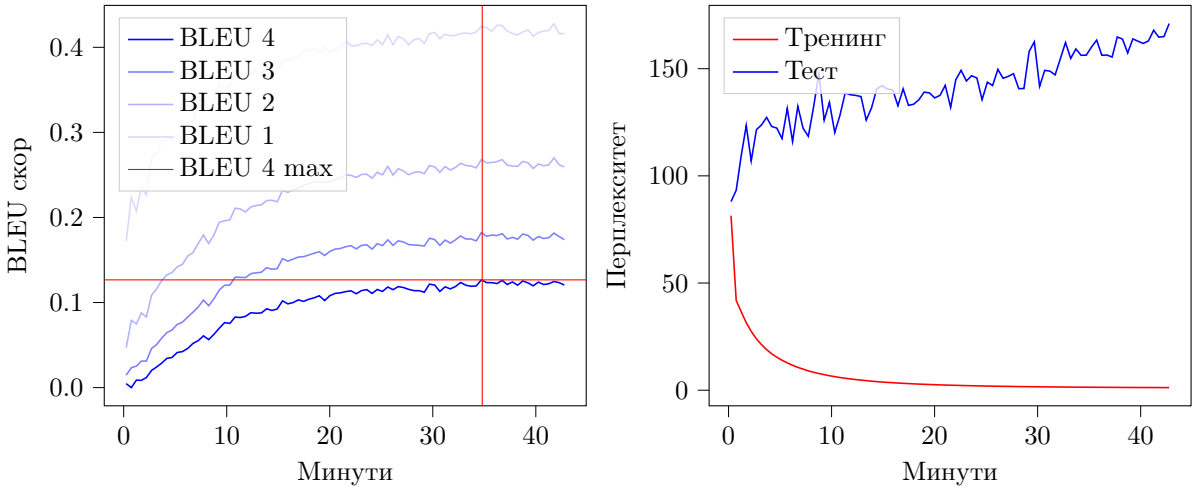
Табела 5.5: Поређење резултата за *SETIMES 40* тест корпус

Листинг 5.4: Узорак *SETIMES 40* тест корпуса преведен експерименталним моделом

```
> proceeds will help fund young athletes .
= prikuplxena sredstva bicxe iskorisxtena za pomocx mladim sportistima .
< prihod cxe pomocxi mladi proizvodxacyi . <EOS>

> other parties advanced their own candidates .
= druge stranke takodxe su istakle svoje kandidate .
< druge stranke imaju sopstvene kandidate . <EOS>

> the eu council hailed the deal as representing a significant step towards
  strengthening bih state institutions and boosting the country s ability to
  meet european standards .
= savet eu pozdravio je dogovor ocenivsxi da predstavlxa znacyajan korak u pravcu
   jacyanxa drzxavnih institucija bih i povecxanxa mogucxnosti zemlxe za
  ispunxavanxe evropskih standarda .
< savet eu pozdravio je sporazum kao znacyajan korak ka jacyanxu bih institucija
  institucija i unapredxivanxe sposobnosti zemlxe da ispune evropske standarde
  standarde . <EOS>

> croatia exports mainly to bosnia and herzegovina slovenia and italy .
= hrvatska izvozi uglavnom u bosnu i hercegovinu sloveniju i italiju .
< hrvatska <UNK> uglavnom bosne i hercegovine slovenije i italije . <EOS>
```

5.3.5 SETIMES 20

Корпус *SETIMES*, сачињен од превода реченица из новинских чланака, пречишћен и ограничен на реченице не дуже од 20 речи. Садржи 10,283 речи и 11,859 парова превода.



(а) BLEU скор током обуке

(б) Перплекситет током обуке

Слика 5.5: Понашање модела током обуке

	BLEU 1 (%)	BLEU 2 (%)	BLEU 3 (%)	BLEU 4 (%)
Google Translate	65.76	52.11	42.67	35.50
Microsoft Translator	74.41	64.56	57.48	**51.85**
Yandex Translate	57.41	42.47	32.60	25.23
Експериментални модел	42.45	26.80	18.22	12.66

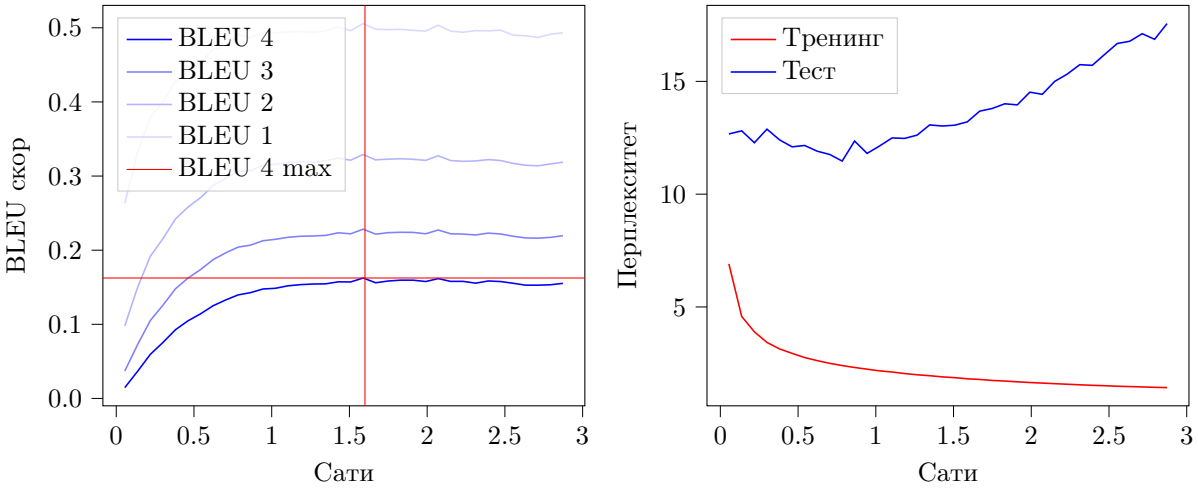
Табела 5.6: Поређење резултата за *SETIMES 20* тест корпус

Листинг 5.5: Узорак *SETIMES 20* тест корпуса преведен експерименталним моделом

```
1  > one condition for bih s eu accession progress is constitutional reform to
       improve the efficiency of state institutions .
2  = jedan od uslova za napredak bih u pridruzxivanxu eu je ustavna reforma kako bi
       se unapredila efikasnost drzxavnih institucija .
3  < jedna od klxucynih gradxana bih eu je zbog <UNK> reforme pravosudxa u
       unapredxivanxu vladavine drzxave . <EOS>
4
5  > ozkok concluded his balkan trip with a stop in albania .
6  = ozkok je zavrsxio svoju balkansku turneju posetom albaniji .
7  < on je ipak svoje balkanske posete u albaniji vecx se sa tim krizom . <EOS>
8
9  > bush reaffirmed us support for turkey s eu bid as well as for its economic
       reforms .
10 = busx je ponovio podrsxku sad kandidaturi turske za prijem u eu kao i ekonomskim
       reformama zemlxe .
11 < busx je potvrdio podrsxku nastojanxima turske za prijem u eu kao i po
       ekonomskom reformama u zemlxi . <EOS>
12
13 > it should not be like that .
14 = to ne bi trebalo da bude tako .
15 < bi to trebalo da bude tako da bude i to desi . <EOS>
```

5.3.6 QED 40

Корпус *QED*, настао упаривањем титлова едукативних видео садржаја, пречишћен и ограничен на реченице не дуже од 40 речи. Садржи 26,792 речи и 83,246 парова превода.



(а) BLEU скор током обуке

(б) Перплекситет током обуке

Слика 5.6: Понашање модела током обуке

	BLEU 1 (%)	BLEU 2 (%)	BLEU 3 (%)	BLEU 4 (%)
Google Translate	60.93	44.93	34.69	**27.11**
Microsoft Translator	57.15	41.00	31.05	24.00
Yandex Translate	51.77	35.37	25.79	19.26
Експериментални модел	50.57	32.91	22.84	16.25

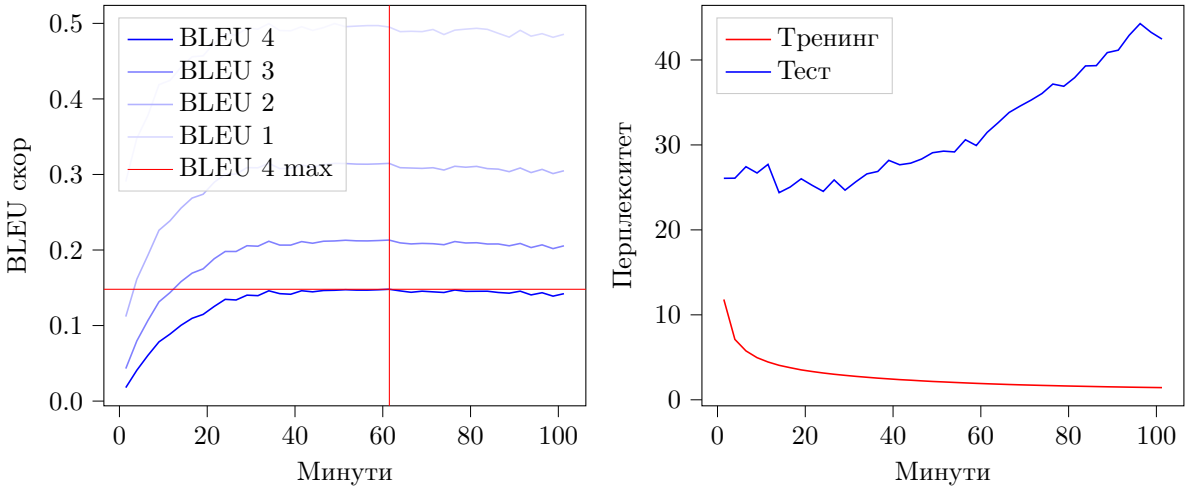
Табела 5.7: Поређење резултата за *QED 40* тест корпус

Листинг 5.6: Узорак *QED 40* тест корпуса преведен експерименталним моделом

```
> now that can work well for an individual learner and i encourage you to do it
    with your kids but i also encourage everyone in the audience to do it
    yourself .
= to mozxe da bude dobro za pojedinca i pozivam vas prvo da radite ovo sa svojom
    decom ali i ohrabrujem sve u publici da probate i vi .
< to mozxe da funkcionisxe dobro za druge vesxtine i vas ohrabrujem vas da
    uradite u vasxem deci ali takodxe i ja volim da u publici to bi bilo . <EOS>

> ah it s okay .
= oh u redu je .
< u redu je . <EOS>

> it s concern with how we get information and how we gather the news .
= briga o tome kako dobijamo informacije i kako skuplxamo vesti .
< to je nesxto kako dobijamo informacije i kako se ophodimo . <EOS>

> and see what happens to your perception .
= i pogledajte sxta se desxava vasxoj percepciji .
< vidite sxta se desxava vasxem glavi . <EOS>
```

5.3.7 QED 20

Корпус *QED*, настао упаривањем титлова едукативних видео садржаја, пречишћен и ограничен на реченице не дуже од 20 речи. Садржи 20,804 речи и 69,581 парова превода.



(а) BLEU скор током обуке

(б) Перплекситет током обуке

Слика 5.7: Понашање модела током обуке

	BLEU 1 (%)	BLEU 2 (%)	BLEU 3 (%)	BLEU 4 (%)
Google Translate	62.12	45.92	35.72	**28.23**
Microsoft Translator	58.16	41.88	31.84	24.61
Yandex Translate	52.34	35.89	26.25	19.67
Експериментални модел	49.52	31.47	21.31	14.80

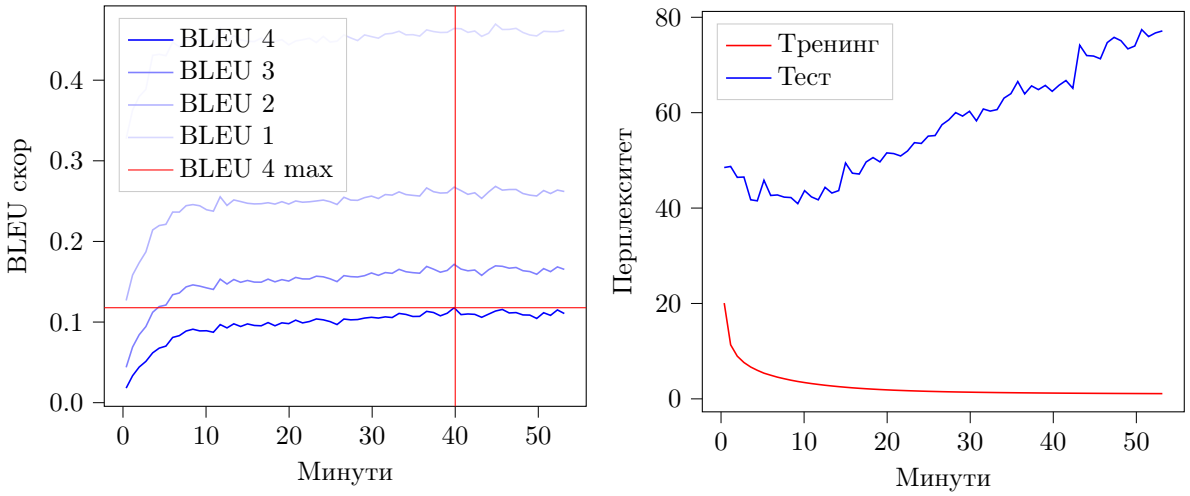
Табела 5.8: Поређење резултата за *QED 20* тест корпус

Листинг 5.7: Узорак *QED 20* тест корпуса преведен експерименталним моделом

```
> but let s say x is how old he is right now .
= ali recimo h je koliko je star sada .
< ali recimo da je h koliko je stara sada . <EOS>

> do you have <UNK> ?
= imasx li <UNK> <UNK> ?
< imate <UNK> ? <EOS>

> it depends on how much energy you have that day so on and so forth .
= zavisi od toga koliko energije imate taj dan i tako dalxe i tako dalxe .
< zavisi od toga koliko energije imate dan tako da se dan i dalxe . <EOS>

> alright so i encouraged you to try this on your own .
= dobro dakle izazivam vas da uradite ovo samostalno .
< u redu to sam vas ohrabrujem da pokusxate ovo da probate . <EOS>
```

5.3.8 QED 10

Корпус *QED*, настао упаривањем титлова едукативних видео садржаја, пречишћен и ограничен на реченице не дуже од 10 речи. Садржи 9,939 речи и 38,042 парова превода.



(а) BLEU скор током обуке

(б) Перплекситет током обуке

Слика 5.8: Понашање модела током обуке

	BLEU 1 (%)	BLEU 2 (%)	BLEU 3 (%)	BLEU 4 (%)
Google Translate	64.19	47.11	36.46	**28.67**
Microsoft Translator	60.94	43.56	33.26	25.90
Yandex Translate	53.09	35.49	25.53	18.77
Експериментални модел	46.42	26.73	17.16	11.78

Табела 5.9: Поређење резултата за *QED 10* тест корпус

Листинг 5.8: Узорак *QED 10* тест корпуса преведен експерименталним моделом

```
> i heard him but i wasn t listening .
= cyuo sam ga ali nisam slusxao .
< rekla sam mu da sam nisam visxe . <EOS>

> yeah give me the torch .
= daj mi <UNK> lampu .
< da mi ipak <UNK> odgovor . <EOS>

> but i think it could .
= ali mislim da bi moglo .
< ali ja mislim da je to jednostavno mogucxe . <EOS>

> now why does stuff like that work ?
= sada zbog cyega takve stvari rade ?
< zasxto to tako funkcionisxe ? <EOS>
```

5.3.9 Закључци

На основу добијених резултата експерименталног модела, јасно се може закључити да квалитет превода не опада повећавањем броја речи у реченици, за шта је заслужан пре свега attention механизам. Штавише, дешава се супротна ствар, што потврђује да перформансе модела јако зависе од величине корпуса.

Међутим, ако се посматрају само скупови са лимитом дужине од 40 речи, уочава се да су резултати обрнуто пропорционални величини корпуса. Узрок се може тражити у величини речника, али ипак, пре свега, у сложености и квалитету превода на коме је модел трениран. Исто важи и за моделе са којима је вршено поређење, са изузетком *Microsoft Translator*-а у једном специфичном случају.

За корпус *Tatoeba 40*, резултат експерименталног модела је веома добар. Може се видети да он постиже тек нешто лошије перформансе од *Google Translate*-а, а да је бољи од друга два конкурента, што је приказано у табели 5.2. На овакав резултат утиче пре свега једноставност и одсуство грешака у ручно написаном *Tatoeba* корпусу. Време потребно за обуку је веома кратко, због мале количине података.

Приметно је да *Google Translate* има најмање варијације у квалитету превода у односу на корпус на коме је тестиран. За *SETIMES* корпусе његови резултати остају константно добри, као и код *Tatoeba* корпуса, без обзира на њихову очигледну разлику у сложености и величини. Код *QED* корпуса резултати су ипак значајно лошији.

Microsoft Translator не заостаје много за *Google Translate*-ом, док у случају *SETIMES* корпуса даје изузетно добар резултат, практично упоредив са врхунским преводом човека преводиоца. Оваква ситуација може се објаснити само чињеницом да је *SETIMES* корпус употребљен за тренирање модела који користи овај сервис и да су добијени резултати уствари BLEU скор његовог тренинг, а не тест скупа.

Yandex Translate сервис значајно заостаје у односу на друга два конкурента, али судећи према томе да на својој веб страници по обављеном преводу кориснику препоручује управо *Google Translate* и *Microsoft Translator*, може се рећи да руска компнија још увек нема намеру да се озбиљно такмичи у овом сегменту.

Из наведеног се може закључити да је експериментални модел успео да озбиљније парира конкурентима само у случају *Tatoeba* корпуса. Разлози за то могу бити недовољна количина података у *SETIMES* корпусима и, имајући у виду резултате других машинских преводилаца, још увек значајна количина грешака нарочито у *QED* корпусима, коју до сада коришћене методе нису успеле довољно да умање.

Што се времена потребног за обуку тиче, оно се повећава заједно са величином корпуса на ком се модел тренира. Ипак, није правило да ће већи корпус изискивати више времена. *QED 40* који је знатно већи од *SETIMES 40* завршава обуку за краће време, што се може оправдати ранијом стагнацијом BLEU скора, која је условљена мањим квалитетом самог корпуса. С обзиром на то да је најдужа обука трајала нешто више од $4h$, може се донети закључак да је експериментални модел испунио захтев који се односи на добре перформансе током процеса тренирања.

У наставку, имајући у виду претходно добијене резултате, биће извршен и додатни експеримент у коме корпуси неће бити дељени према броју речи у реченици, а биће додатно пречишћени новом финијом методом, за коју су стечени потребни услови. Такође, биће извршено и обучавање модела на комбинованом корпусу који је настао спајањем сва три до сада коришћена корпуса, како би се стекао увид у понашање модела при тренирању над већим скупом података.

5.4 NMT пречишћавање

Методе за пречишћавање које су до сада коришћене, имају ефекта и на више различитих начина у стању су да открију и уклоне проблематичне парове превода. Међутим, оне ипак не решавају у довољној мери проблем лоше преведеног или лоше упареног превода.

Прилично поуздан начин да се аутоматски провери прихватљивост превода је његово поређење са резултатом који даје неки MT модел. Поређење би се могло вршити функцијом *sentence_bleu* из *nltk.translate.bleu_score* библиотеке. *SmoothingFunction* метода 4 [13], обезбедила би сигурност краћих превода. Ако би лимит за прихватање био 0.2, постојала би веома сигурна шанса да лоше упарени преводи буду уклоњени, солидна шанса да добро упарени, а ипак лоши преводи, такође не прођу проверу, док би опасност за уклањање добрих парова превода била мала.

Експериментални модели, са постигнутим резултатима, могу бити употребљени за MT пречишћавање, али било би проблема, пре свега, због недовољно великог речника којим они располажу. Због тога би употреба *Google Translate* сервиса, који се генерално најбоље показао у претходним експериментима, за ову намену била адекватнија.

Проблем настаје када се узму у обир лимити које сервис поставља у виду величине докумената који се могу превести, а које су у овом случају 9 пута веће од оних потребних за превођење тест корпуса зарад рачунања BLEU скорова. Компанија нуди и напредније и компликованије опције које нису бесплатне, док међутим пројекат *DocTranslator* [19] базиран на *Google Translate* сервису нуди могућност бесплатног превођења нешто већих докумената. Уз поделу енглеског дела корпуса на мање целине и уз довољно стрпљења овај сервис може се успешно применити у датој ситуацији.

Скрипта дата у листингу 5.9 дели текстуални фајл на делове не веће од $256KB$ без сечења линија.

Листинг 5.9: corpusSpliter.sh

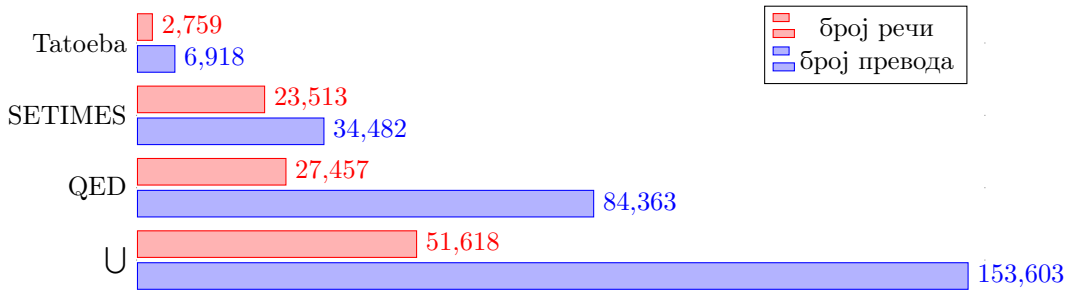
```
#!/bin/bash
split -d -C 256KB --suffix-length=3 --additional-suffix=.txt $1
```

На основу добијених превода могу се креирати хипотезе, а њих је даље могуће тестирати на преводима датим у корпусу који представљају референце. Тестирање се може спровести скриптом датом у листингу 5.10.

Листинг 5.10: checkBLEU.py

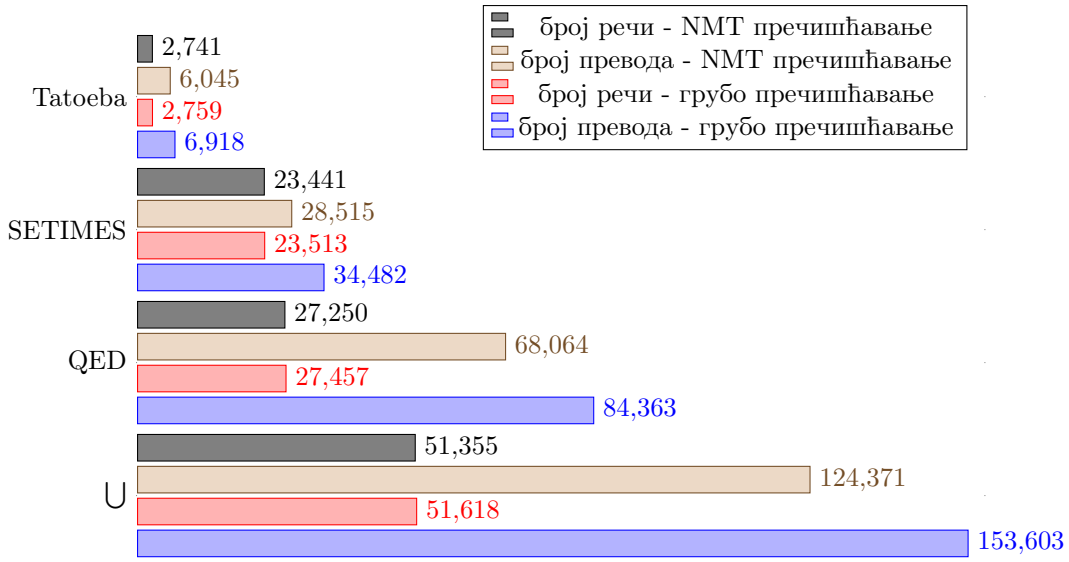
```
def checkBLEU(reference, hypothese)

    s = SmoothingFunction().method4
    try:
        if sentence_bleu(reference, hypothese, smoothing_function=s) > 0.2 :
            return True

    except ZeroDivisionError:
        print('ZeroDivisionError!')

    return False
```

Методе коришћене у поглављу 4 могу се сматрати грубим пречишћавањем. На основу закључка добијеног из експеримента, нема потребе за лимитирањем дужине реченице. Када се спроведе грубо пречишћавање три расположива корпуса, као и њихове уније, добија се резултат дат на слици 5.9.



Слика 5.9: Број речи и превода у грубо пречишћеним корпусима

На грубо пречишћеним корпусима примењено је NMT пречишћавање по претходно описаном поступку. Резултати су дати на слици 5.10.



Слика 5.10: Број речи и превода после NMT пречишћавања

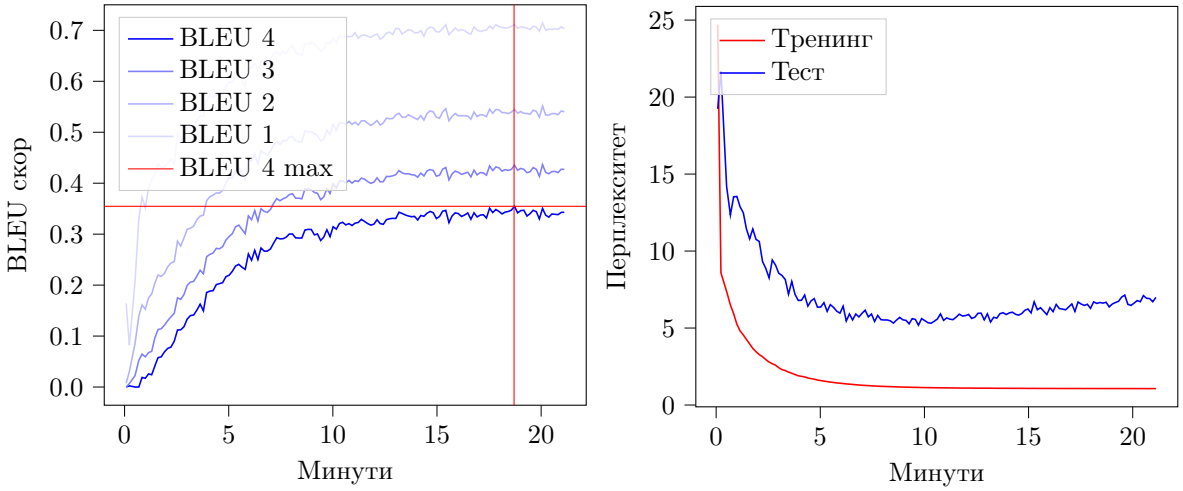
Скрипте коришћене за NMT пречишћавање налазе се на јавном репозиторијуму [2].

[2] https://gitlab.com/dragutinostojic/nmtsren/-/tree/master/clearCorpus

5.5 Резултати

5.5.1 Tatoeba

Наменски писан корпус *Tatoeba*, пречишћен прво грубом, потом NMT методом. Садржи 2,741 речи и 6,045 парова превода.



(а) BLEU скор током обуке

(б) Перплекситет током обуке

Слика 5.11: Понашање модела током обуке

	BLEU 1 (%)	BLEU 2 (%)	BLEU 3 (%)	BLEU 4 (%)
Google Translate	71.13	57.84	46.42	**38.15**
Microsoft Translator	68.09	51.41	41.24	33.82
Yandex Translate	56.40	37.93	27.62	20.52
Експериментални модел	71.11	54.50	43.54	35.47

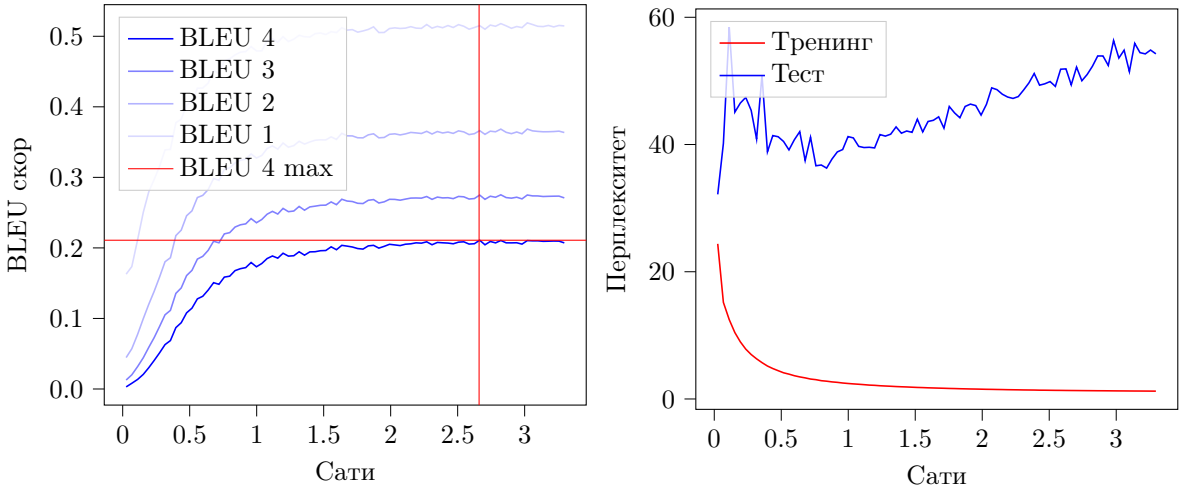
Табела 5.10: Поређење резултата за *Tatoeba* тест корпус

Листинг 5.11: Узорак *Tatoeba* тест корпуса преведен експерименталним моделом

```
> what i saw was depressing .
= ono sxto sam videla je bilo <UNK> .
< ono sxto sam video je bilo <UNK> . <EOS>

> tom used to be scared of snakes .
= tom se ranije plasxio zmija .
< tom se nekada plasxio zmija . <EOS>

> tom looked very puzzled .
= tom je izgledao jako zbunxeno .
< tom je delovao veoma zbunxeno . <EOS>

> can you explain to me how i can get to the airport ?
= mozxete li da mi objasnite kako da stignem do aerodroma ?
< da li mozxete da mi objasnite kako da stignem do aerodroma ? <EOS>
```

5.5.2 SETIMES

Корпус *SETIMES*, сачињен од превода реченица из новинских чланака, пречишћен прво грубом, а затим NMT методом. Садржи 23,441 речи и 28,515 парова превода.



(а) BLEU скор током обуке

(б) Перплексиетет током обуке

Слика 5.12: Понашање модела током обуке

	BLEU 1 (%)	BLEU 2 (%)	BLEU 3 (%)	BLEU 4 (%)
Google Translate	67.74	54.99	45.75	38.49
Microsoft Translator	76.48	67.66	60.94	**55.38**
Yandex Translate	58.21	43.44	33.48	26.11
Експериментални модел	51.51	36.63	27.49	21.09

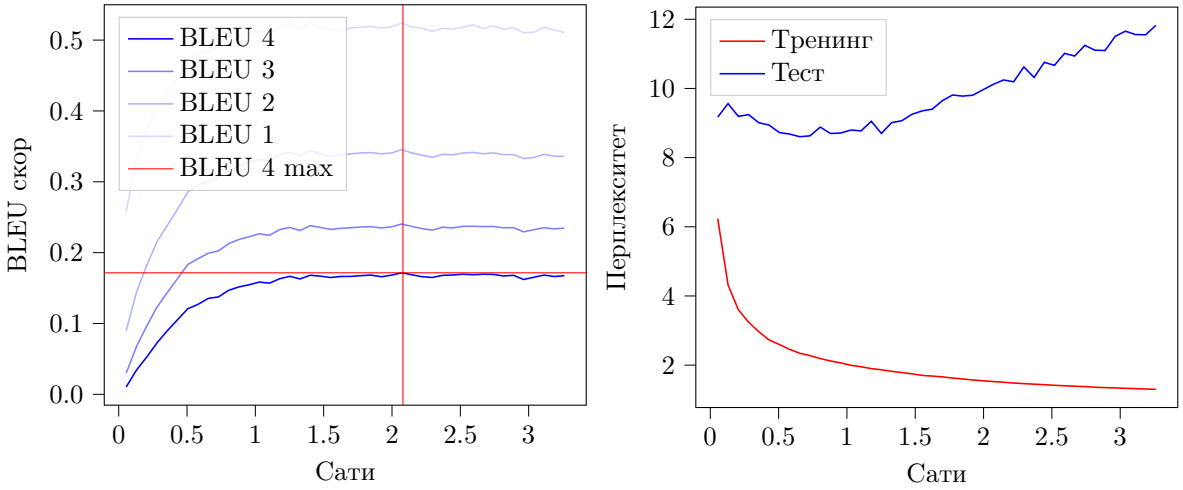
Табела 5.11: Поређење резултата за *SETIMES* тест корпус

Листинг 5.12: Узорак *SETIMES* тест корпуса преведен експерименталним моделом

```
> former bosnian serb leader radovan karadzic and his military commander ratko
  mladic top the list of four indictees still sought by the international
  criminal tribunal for the former yugoslavia .
= bivsxi lider bosanskih srba radovan karadyicx i nxegov vojni komandant ratko
  mladicx nalaze se na vrhu liste cyetvorice optuzxenika koje josx uvek trazxi
  medxunarodni krivicyni sud za bivsxu jugoslaviju .
< bivsxi lider bosanskih srba radovan karadyicx i nxegov vojni komandant ratko
  mladicx za se na vrhu protiv cyetvorice optuzxenih josx uvek trazxi da trazxi
   medxunarodni sud za bivsxu jugoslaviju . <EOS>

> the <UNK> tone of the report was in keeping with predictions .
= upozoravajucxi ton izvesxtaja bio je u skladu sa prognozama .
< <UNK> izvesxtajem u znak uz <UNK> iz vlasti . <EOS>

> the full impact of the crisis has yet to be measured .
= potpuni uticaj krize tek treba da bude procenxen .
< potpuna uticaj krize josx uvek nije odredxen . <EOS>

> many ministries received less money than last year .
= mnoga ministarstva dobila su manxe novca nego prosxle godine .
< mnogi su dobili dobili manxe od prosxle godine . <EOS>
```

5.5.3 QED

Корпус *QED*, настао упаривањем титлова едукативних видео садржаја, пречишћен прво грубом, а затим NMT методом. Садржи 27,250 речи и 68,064 парова превода.



(а) BLEU скор током обуке

(б) Перплекситет током обуке

Слика 5.13: Понашање модела током обуке

	BLEU 1 (%)	BLEU 2 (%)	BLEU 3 (%)	BLEU 4 (%)
Google Translate	64.93	49.24	38.79	**30.91**
Microsoft Translator	59.61	43.76	33.54	26.03
Yandex Translate	54.12	37.77	27.89	20.98
Експериментални модел	52.42	34.54	24.04	17.16

Табела 5.12: Поређење резултата за *QED* тест корпус

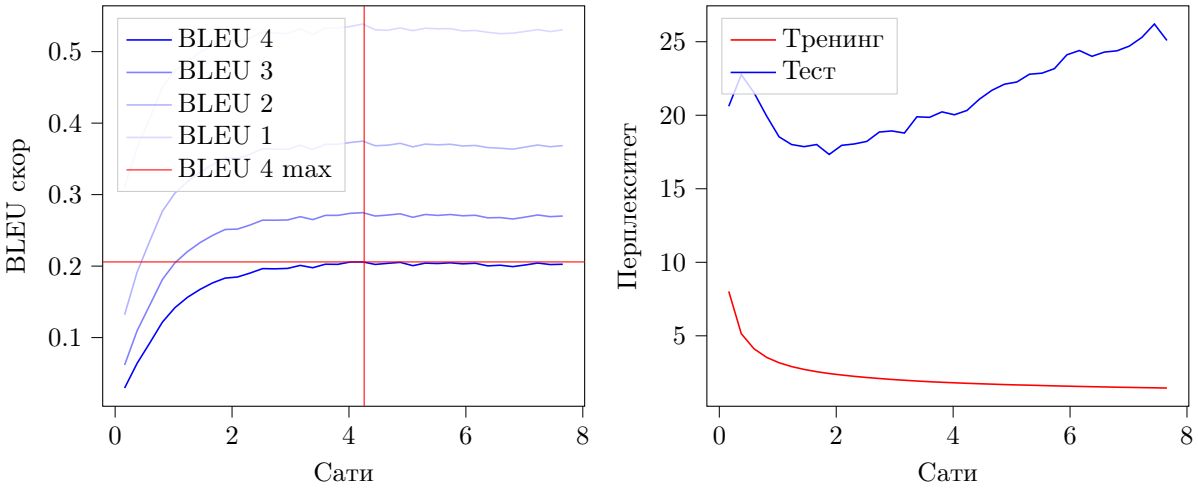
Листинг 5.13: Узорак *QED* тест корпуса преведен експерименталним моделом

```
> let s solve this equation .
= hajde da resximo ovu jednacyinu .
< hajde da resximo ovu jednacyinu . <EOS>

> of course it s this big .
= svakako on je ovako velik .
< naravno je to ovaj veliki . <EOS>

> now i don t know if this is the devil or the angel sitting on our conscience
  sitting on television s shoulders but i do know that i absolutely love this
  image .
= ne znam da li je taj koji nam sedi na savesti i na ramenima televizije dxavo
  ili andxeo ali znam da apsolutno obozxavam ovu sliku .
< ne znam da li je ovo <UNK> ili pravila koji sedi na svim licu ali ja znam da ja
   apsolutno apsolutno obozxavam ovu sliku . <EOS>

> see what you can pick up from this .
= vidite sxta mozxete uocyiti ovde .
< vidite sxta mozxete da uradite iz ovoga . <EOS>
```

5.5.4 Унија корпуса

Унија оригиналних *Tatoeba*, *SETIMES* и *QED* корпуса, пречишћена прво грубом, а затим NMT методом. Садржи 51,355 речи и 124,371 парова превода.

Са датим хиперпараметрима у табели 5.1 и величином пречишћене уније корпуса, која се може видети на графикону 5.10, 12*GB* меморије које поседује *TITAN Xp* картица није довољно за обуку. Због тога величина batch-a, у овом експерименту, смањена је на 64.



(а) BLEU скор током обуке

(б) Перплекситет током обуке

Слика 5.14: Понашање модела током обуке

	BLEU 1 (%)	BLEU 2 (%)	BLEU 3 (%)	BLEU 4 (%)
Google Translate	66.85	52.67	42.90	35.45
Microsoft Translator	67.16	54.79	46.50	**40.28**
Yandex Translate	54.47	39.90	30.11	23.13
Експериментални модел	53.86	37.48	27.48	20.58

Табела 5.13: Поређење резултата за $\bigcup$ тест корпус

Листинг 5.14: Узорак $\bigcup$ тест корпуса преведен експерименталним моделом

```
> well let s try to understand why exactly it worked .
= hajde da pokusxamo da razumemo zasxto je sve to funkcionisalo .
< pa hajde da probamo zasxto je tacyno uspelo . <EOS>

> what did you try to do ?
= sxta si pokusxao da uradisx ?
< sxta si pokusxao da uradisx ? <EOS>

> same as the other side .
= isto kao na drugoj strani .
< isto kao i sa druge strane . <EOS>

> i am sorry i cannot show you my face because if i do the bad guys will come for
    me .
= zxao mi je ali ne mogu da vam pokazxem svoje lice jer ako ga pokazxem losxi
  momci cxe docxi po mene .
< zxao mi je sxto ne mogu da vam pokazxem lice jer ako ja to uradim losx momci .
  <EOS>
```

5.5.5 Закључци

На основу добијених резултата, приметно је да NMT метода пречишћавања поправља BLEU скор. Резултати машинских преводилаца су бољи.

Експериментални модел и *Google Translate* производе јако квалитетне преводе када је у питању *Tatoeba* корпус. Код *SETIMES* корпуса, експериментални модел сада има резултат већи од 20, што се, условно речено, може сматрати прихватљивим. Код *QED* корпуса сви модели показују лошије резултате него иначе, што имплицира да и даље постоје проблеми са његовим садржајем, који би се потенцијално могли отклонити повећавањем границе за одстрањивање у NMT методи пречишћавања. Даље, код уније корпуса, машински преводиоци дају боље резултате него код *QED* корпуса.

Највећи скок забележио је *Google Translate* сервис и такав резултат може оправдати чињеница да су уклоњени баш они преводи који су њему конкретно правили највише проблема. Међутим, чињеница да је експериментални модел, као и остали машински преводиоци, направио бољи скор након пречишћавања, говори да су одстрањени баш они парови који су и објективно били проблематични, врло вероватно због лошег превода или лошег упаривања.

Добијени модел, трениран над унијом примењиваних корпуса, ипак не постиже довољно добре резултате да би могао да се користи у пракси. Међутим, он у тренутном стању садржи довољно квалитетан речник и карактеристике да би могао да послужи као алат за NMT пречишћавање других корпуса.

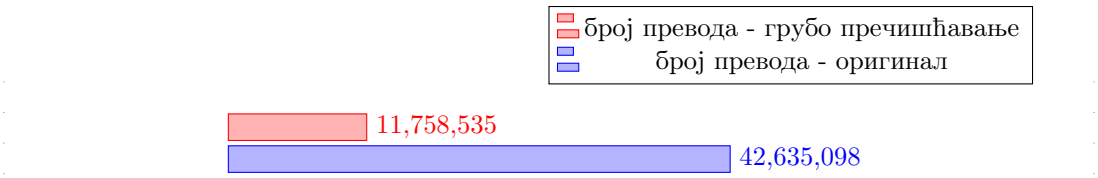
5.6 OpenSubtitles

Тренутно највећи доступан двојезични корпус који садржи и српски језик сачињен је од титлова из *OpenSubtitles* архиве. Овај корпус је у старту одбачен у поглављу 4 због великих проблема са упаривањем.

Пречишћавање овог корпуса NMT методом отвара могућности за његову употребу у тренирању неуронских машинских преводилаца. Коришћење сервиса које нуде велике компаније за ову намену, што је до сада био случај са *Google Translate*-ом, тешко је изводљиво због велике количине података коју је потребно превести. Чак и уз куповину посебних лиценци, постављени лимити су значајно испод неопходних за обављање овог задатка у разумном времену.

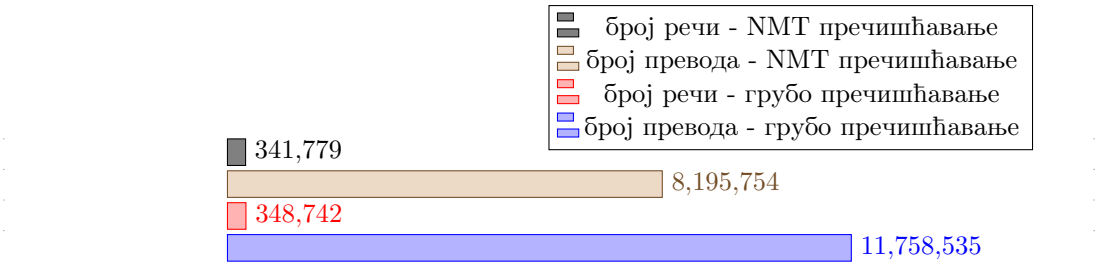
Међутим, трениран над унијом корпуса, екпериментални модел, иако можда недовољно добар за практичну употребу, у поређењу са најуспешнијим конкурентима, може користити као сасвим адекватан алат за NMT пречишћавање других енглеско-српских корпуса, међу којима је и *OpenSubtitles*.

Грубим пречишћавањем добијају се резултати дати на слици 5.15.



Слика 5.15: Број превода пре и после грубог пречишћавања

Употребом скрипте дате у листингу 5.10 на основу превода експерименталног модела тренираног на $\bigcup$ корпусу добијени су резултати дати на слици 5.16.



Слика 5.16: Број речи и превода после NMT пречишћавања

Због хардверских ограничења, током писања овог рада није било могуће ефикасно тренирати експериментални модел на овом корпусу. На основу досадашњих закључака, његове карактеристике које укључују једноставан језик и велику количину превода, уз потврђену успешност NMT пречишћавања, наговештавају добар резултат.

Као знатно већи од свих корпуса коришћених у овом раду, *OpenSubtitles* представља веома битан ресурс за даља истраживања, која би могла дати додатне одговоре који се тичу архитектуре експерименталног модела и NMT метода пречишћавања.

Глава 6

Закључак

Неуронске мреже са Енкодер-Декодер архитектуром испоставиле су се као врло моћан алат за машинско превођење. У овом раду представљен је пример употребе тог приступа за превођење са енглеског на српски језик, као и поређење перформанси добијеног модела са познатим комерцијалним машинским преводиоцима.

Имплементација је укључила различите технике за побољшавање квалитета превода, као и перформанси везаних за сам процес обуке. Главне предности конструисаног модела су једноставност и мала хардверска захтевност, и оне, као такве, остављају пуно простора за даљи развој. Суштински недостатак конструисаног модела је генерално мањи квалитет превода у односу на познате конкуренте. Разлози за то могу се тражити пре свега у недовољно великом броју и величини доступних и адекватних корпуса у време писања овог рада. Са друге стране, прикупљени су и анализирани вероватно сви доступни енглеско-српски корпуси.

Током истраживања примећено је више занимљивих резултата. Процес подешавања хиперпараметара потврдио је открића ранијих истраживања у овој области, при чему су одређене и неке специфичне вредности за дати модел и његову примену. Такође, спроведени експерименти показују да ограничавање дужине реченице у корпусу не утиче позитивно на BLEU скор, што доприноси становишту да дужина реченице не може бити мера њене комплексности. Уведен је оригинални начин грубог пречишћавања корпуса, који је посебно осмишљен за енглеско-српске паралелне корпусе, као и NMT пречишћавање које се може универзално користити. Показало се и да пречишћавање NMT моделом тренираним на мањем корпусу даје јако добре резултате.

Необична појава која се може приметити је и несразмеран однос добијених вредности перплекситета и BLEU скора у извршеним експериментима. Он је приметан мање или више на графиконима који прате ток тренирања свих представљених модела. За овај феномен одговоран је пре свега BLEU скор, који поред велике примене и опште прихваћености има и одређене недостатке [81]. У спроведеним експериментима тежило се што већем BLEU скору, док је перплекситет жртвован зарад тог циља. Рад који представља доказ ефикасности овог приступа објављен је у завршним фазама писања овог мастер рада [59].

Сам концепт Енкодер-Декодер модела, иако и у овом облику даје значајна побољшања у односу на претходно коришћене системе за статистичко машинско превођење, оставља пуно простора за унапређење, и као нова технологија има велики потенцијал. У вези с тим, проширивање конструисаног модела постојећим технологијама, од којих неке нису имплементиране, и оним технологијама које су откривене у току писања овог рада, добар су начин за даље унапређење. Међутим, највећа шанса за укупно побољшавање квалитета превода могла би лежати у унапређењу механизама за пречишћавање корпуса. Захваљујући пре свега NMT фази, у зрелој верзији такав систем можда би могао да оствари довољну прецизност за аутоматску паралелизацију материјала као што су званични преводи књига, чиме би се трајно решио проблем недостатка расположивих корпуса.

Значајно је напоменути да су 2017. године Васвани *(енгл. Ashish Vaswani)* и други увели нови приступ у машинском превођењу и NLP-у уопште који носи назив *Transformer model* [87]. Овај принцип се значајно разликује од NMT методе коришћене у овом раду и има велики потенцијал да у будућности постане стандард у овој области. Главне његове карактеристике су то што не захтева унос секвенце у тачно одређеном редоследу, као и то што отвара много могућности за паралелизацију. У даљим истраживањима биће вршени експерименти и са овом архитектуром неуронских мрежа.

На крају, може се донети закључак да је овај рад потврдио учинковитост рекурентних неуронских мрежа у обради природних језика, на јако захтевном примеру машинског превођења. Будућа истраживања биће фокусирана на даље унапређивање модела, његово тренирање на већим корпусима, као и стварање метода за генерисање нових корпуса, као и тестирање нових приступа у машинском превођењу.

Литература

[1] Ahmed Abdelali и др. „The AMARA Corpus: Building Parallel Language Resources for the Educational Domain". јануар 2014. DOI: 10.13140/2.1.1806.2406.

[2] M. Akimoto и др. *Language Change and Variation from Old English to Late Modern English: A Festschrift for Minoji Akimoto*. Linguistic Insights: Studies in Language and Communication. Peter Lang, 2010., стр. 21–34. ISBN: 9783034303729.

[3] Antonios Anastasopoulos. „An Analysis of Source-Side Grammatical Errors in NMT". *Proc. BlackboxNLP*. 2019.

[4] Mihael Arčan, Maja Popovic и Paul Buitelaar. „Asistent – A Machine Translation System for Slovene, Serbian and Croatian". септембар 2016.

[5] Dzmitry Bahdanau, Kyunghyun Cho и Yoshua Bengio. „Neural Machine Translation by Jointly Learning to Align and Translate". *CoRR* abs/1409.0473 (2015.).

[6] *Bank of English*. University of Birmingham. 1994. URL: http://cqpweb.bham.ac.uk.

[7] Yoshua Bengio, Patrice Simard и Paolo Frasconi. „Learning long-term dependencies with gradient descent is difficult". *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council* 5 (фебруар 1994.), стр. 157–66. DOI: 10.1109/72.279181.

[8] Zhu Qi-bo. *Guangzhou Petroleum English Corpus*. Guangzhou Training College of the Chinese Petroleum University. 1987.

[9] Denny Britz и др. „Massive exploration of neural machine translation architectures". (2017.). arXiv: 1703.03906.

[10] P. Brown и др. „A statistical approach to language translation". (август 1988.), стр. 71–76. DOI: 10.3115/991635.991651.

[11] J. Brownlee. *Deep Learning for Natural Language Processing: Develop Deep Learning Models for your Natural Language Problems*. Machine Learning Mastery, 2017. URL: https://books.google.rs/books?id=%5C_pmoDwAAQBAJ.

[12] R. Carter. *Linguistics and the Teacher*. Linguistics and the Teacher v. 112. Routledge, 2012. ISBN: 9780415694261.

[13] Boxing Chen и Colin Cherry. „A Systematic Comparison of Smoothing Techniques for Sentence-Level BLEU". *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, јун 2014., стр. 362–367. DOI: 10.3115/v1/W14-3346.

[14] Kyunghyun Cho и др. „Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, октобар 2014., стр. 1724–1734. DOI: 10.3115/v1/D14-1179.

[15] N. Chomsky. *Syntactic Structures*. Janua linguarum: Series minor. Mouton, 1957. URL: https://books.google.rs/books?id=5gJZGD4YXF0C.

[16] Christos Christodoulopoulos и Mark Steedman. „A massively parallel corpus: the Bible in 100 languages". *Language Resources and Evaluation* 49 (јун 2014.), стр. 1–21. DOI: 10.1007/s10579-014-9287-y.

[17] BNC Consortium. *British National Corpus*. 1994. URL: http://natcorp.ox.ac.uk.

[18] H. B. Curry. „The Method of Steepest Descent for Non-Linear Minimization Problems". *Quarterly of Applied Mathematics* 2.3 (1944.), стр. 258–261. ISSN: 0033569X, 15524485.

[19] *DocTranslator*. URL: https://www.onlinedoctranslator.com/ (посећено 30. 5. 2020.).

[20] T. Etchegoyhen и др. „Machine Translation for Subtitling: A Large-Scale Evaluation". *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association, 2014., стр. 46–53. URL: http://lrec-conf.org/proceedings/lrec2014/pdf/463_Paper.pdf.

[21] Felix Gers и Jurgen Schmidhuber. „Recurrent nets that time and count". св. 3. фебруар 2000., 189–194 vol.3. ISBN: 0-7695-0619-4. DOI: 10.1109/IJCNN.2000.861302.

[22] Gabriel Grand и др. „Semantic projection: recovering human knowledge of multiple, distinct object features from word embeddings". (2018.). arXiv: 1802.01241.

[23] Klaus Greff и др. „LSTM: A search space odyssey". *IEEE transactions on neural networks and learning systems* 28 (март 2015.). DOI: 10.1109/TNNLS.2016.2582924.

[24] Michael Hancher. *English Poetry Full-Text Database*. Department of English, University of Minnesota. 1995. URL: http://mh.cla.umn.edu/websites.html.

[25] Zellig S. Harris. „Distributional structure". *Word* 10.2-3 (1954.), стр. 146–162.

[26] Michael Hart. *Project Gutenberg*. 1971. URL: https://www.gutenberg.org/.

[27] Sepp Hochreiter. „Untersuchungen zu dynamischen neuronalen Netzen". немачки. (1991.).

[28] Sepp Hochreiter и Jürgen Schmidhuber. „Long Short-Term Memory". *Neural Computation* 9.8 (1997.), стр. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[29] W.J. Hutchins. *Early Years in Machine Translation: Memoirs and biographies of pioneers*. Studies in the History of the Language Sciences. John Benjamins Publishing Company, 2000. ISBN: 9789027283719.

[30] *International Corpus of English*. University of Zurich. 1990. URL: http://ice-corpora.net/ice/index.html/.

[31] Herbert Jaeger и Harald Haas. „Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication". *Science (New York, N.Y.)* 304 (мај 2004.), стр. 78–80. DOI: 10.1126/science.1091277.

[32] Stig Johansson и др. *The English-Norwegian Parallel Corpus*. Dept. of British and American Studies, University of Oslo. 1997. URL: http://www.hf.uio.no/ilos/english/services/omc/enpc/.

[33] Rafal Jozefowicz, Wojciech Zaremba и Ilya Sutskever. „An Empirical Exploration of Recurrent Network Architectures". *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015., стр. 2342–2350.

[34] D. Jurafsky и J.H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall series in artificial intelligence. Pearson Prentice Hall, 2009., стр. 177. ISBN: 9780131873216.

[35] Andrej Karpathy. *The Unreasonable Effectiveness of Recurrent Neural Networks*. URL: http://karpathy.github.io/2015/05/21/rnn-effectiveness/ (посећено 14. 9. 2019.).

[36] Philipp Koehn. „Europarl: A parallel corpus for statistical machine translation". *MT summit*. св. 5. Citeseer. 2005., стр. 79–86.

[37] Philipp Koehn и Rebecca Knowles. „Six Challenges for Neural Machine Translation". *Proceedings of the First Workshop on Neural Machine Translation*. Vancouver: Association for Computational Linguistics, август 2017., стр. 28–39. DOI: 10.18653/v1/W17-3204.

[38] Reinhard Köhler, Altmann Gabriel и Genrikhovich Piotrovskiĭ Raĭmond. *Quantitative Linguistics : An International Handbook*. W. de Gruyter, 2005. ISBN: 9783110155785.

[39] Andras Kornai. *Mathematical Linguistics*. Advanced Information and Knowledge Processing. Springer London, 2007. ISBN: 9781846289859.

[40] Dragan Kosovac. *Online prevođenje*. Svet kompjutera. URL: https://www.sk.rs/2012/04/skin06.html (посећено 15. 1. 2020.).

[41] Jan Koutník и др. „A clockwork RNN". *31st International Conference on Machine Learning, ICML 2014* 5 (фебруар 2014.).

[42] Cvetana Krstev и Duško Vitas. „An Aligned English-Serbian Corpus". *ELLSIIR Proceedings (English Language and Literature Studies: Image, Identity, Reality)* 1 (децембар 2009.). ур. N. Tomović и J. Vujić, стр. 495–508.

[43] Henry Kučera. „Computational Analysis of Predicational Structures in English". *Proceedings of the 8th Conference on Computational Linguistics*. COLING '80. Tokyo, Japan: Association for Computational Linguistics, 1980., стр. 32–37. DOI: 10.3115/990174.990181.

[44] Quinn M. Lanners и Thomas Laurent. „Neural Machine Translation". 2019.

[45] J. S. Lario. *Modeos Gramaticales del Inglés*. шпански. Universidad de Granada. 2015.

[46] G. Leech и др. *The Lancaster-Oslo/Bergen Corpus*. Longman Group Limited, the British Academy, Department of British and American Studies, University of Oslo, Norwegian Research Council for Science and the Humanities, Norwegian Computing Centre for the Humanities. 1976. URL: http://helsinki.fi/varieng/CoRD/corpora/LOB/index.html.

[47] Mark Liberman. *Real trends in word and sentence length*. URL: https://languagelog.ldc.upenn.edu/nll/?p=3534 (посећено 12. 2. 2020.).

[48] Pierre Lison и Jörg Tiedemann. „OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles". *LREC*. 2016. URL: http://www.opensubtitles.org/.

[49] Pintu Lohar, Maja Popovic и Andy Way. „Building English-to-Serbian Machine Translation System for IMDb Movie Reviews". август 2019. DOI: 10.18653/v1/W19-3715.

[50] Minh-Thang Luong, Hieu Pham и Christopher D. Manning. „Effective Approaches to Attention-based Neural Machine Translation". (2015.). arXiv: 1508.04025.

[51] N. Ljubesic, P. Bago и D. Boras. „Statistical Machine Translation of Croatian Weather Forecasts: How Much Data Do We Need?": *CIT* 18 (2010.). DOI: 10.2498/cit.1001917.

[52] Nikola Ljubešić. *SETimes – A Parallel Corpus of English and South-East European Languages*. URL: http://nlp.ffzg.hr/resources/corpora/setimes/.

[53] Mirjam Maučec, Janez Brest и Zdravko Kacic. „Slovenian to English machine translation using corpora of different sizes and morpho-syntactic information". (јануар 2006.).

[54] T. Mcenery и A. Wilson. *Corpus Linguistics*. јануар 2001. DOI: 10.1093/oxfordhb/9780199276349.013.0024.

[55] Alice ter Meulen. „Logic and Natural Language". *The Blackwell Guide to Philosophical Logic*. John Wiley & Sons, Ltd, 2017. погл. 20, стр. 461–483. ISBN: 9781405164801. DOI: 10.1002/9781405164801.ch20.

[56] Tomas Mikolov и др. „Efficient Estimation of Word Representations in Vector Space". јануар 2013., стр. 1–12.

[57] Milos Miljanovic. „Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction". *Indian Journal of Computer Science and Engineering* 3 (фебруар 2012.).

[58] M. Nagao и др. „A Machine Translation System from Japanese into English: Another Perspective of MT Systems". *Proceedings of the 8th Conference on Computational Linguistics*. COLING '80. Tokyo, Japan: Association for Computational Linguistics, 1980., стр. 414–423. DOI: 10.3115/990174.990250.

[59] Xuan-Phi Nguyen и др. „Data Diversification: An Elegant Strategy For Neural Machine Translation". (2019.). arXiv: 1911.01986.

[60] Anne O'Keeffe и Michael McCarthy. *The Routledge Handbook of Corpus Linguistics*. Routledge handbooks in applied linguistics Routledge handbooks. Routledge, 2010. ISBN: 9780415464895.

[61] Christopher Olah. *Understanding LSTM Networks*. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (посећено 11. 9. 2019.).

[62] Kishore Papineni и др. „Bleu: a Method for Automatic Evaluation of Machine Translation". *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, јул 2002., стр. 311–318. DOI: 10.3115/1073083.1073135.

[63] Razvan Pascanu, Tomas Mikolov и Yoshua Bengio. „On the Difficulty of Training Recurrent Neural Networks". *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA, USA: JMLR.org, 2013., III–1310–III–1318.

[64] Jeffrey Pennington, Richard Socher и Christoper Manning. „Glove: Global Vectors for Word Representation". св. 14. јануар 2014., стр. 1532–1543. DOI: 10.3115/v1/D14-1162.

[65] Katharine Perera. „The Assessment of Linguistic Difficulty in Reading Material". *Educational Review* 32.2 (1980.), стр. 151–161. DOI: 10.1080/0013191800320204.

[66] Mikael Phi. *Illustrated Guide to Recurrent Neural Networks: Understanding the Intuition*. URL: https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9 (посећено 21. 10. 2019.).

[67] Maja Popovic. „Language-related issues for NMT and PBMT for English–German and English–Serbian". *Machine Translation* (2018.). DOI: 10.1007/s10590-018-9219-5.

[68] Maja Popovic и Mihael Arčan. „Identifying main obstacles for statistical machine translation of morphologically rich South Slavic languages". мај 2015.

[69] Maja Popović и др. „Augmenting a Small Parallel Text with Morpho-Syntactic Language Resources for Serbian-English Statistical Machine Translation". *Proceedings of the ACL Workshop on Building and Using Parallel Texts*. ParaText '05. Ann Arbor, Michigan: Association for Computational Linguistics, 2005., стр. 41–48.

[70] Randolph Quirk. *Survey of English Usage*. University College London. 1959. URL: https://www.ucl.ac.uk/english-usage/.

[71] M. Rissanen. *Helsinki Corpus of English Texts*. The University of Helsinki; The Academy of Finland. 1991. URL: http://helsinki.fi/varieng/CoRD/corpora/HelsinkiCorpus.

[72] Sean Robertson. *Translation with a Sequence to Sequence Network and Attention*. ур. PyTorch. 2018. URL: https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html (посећено 16. 2. 2020.).

[73] Miro Romih и Peter Holozan. „Slovensko-angleški prevajalni sistem". словеначки. *In Proceedings of the 3rd Language Technologies Con-ference (in Slovenian)* (2002.).

[74] Yubin Ruan. *Rnn Immediate Partial Derivative*. URL: https://walkerlala.github.io/archive/rnn-immediate-partial-derivative.html (посећено 4. 11. 2019.).

[75] David E. Rumelhart, Geoffrey E. Hinton и Ronald J. Williams. „Learning representations by back-propagating errors". *Nature* 323 (1986.), стр. 533–536.

[76] Erik Schils и Pieter de Haan. „Characteristics of Sentence Length in Running Text". *Literary and Linguistic Computing* 8.1 (јануар 1993.), стр. 20–26. ISSN: 0268-1145.

[77] M. Schuster и K. K. Paliwal. „Bidirectional recurrent neural networks". *IEEE Transactions on Signal Processing* 45.11 (1997.), стр. 2673–2681.

[78] S. V. Shastri. *Kolhapur Corpus of Indian English*. Department Of English Shivaji University. 1986. URL: http://ice-corpora.net/ice/index.html/.

[79] Ilya Sutskever, Oriol Vinyals и Quoc Le. „Sequence to Sequence Learning with Neural Networks". *Advances in Neural Information Processing Systems* 4 (септембар 2014.).

[80] Jan Svartvik. *The London-Lund corpus of spoken English : description and research*. Lund, Sweden : Lund University Press ; Bromley, England : Chartwell-Bratt, 1990. ISBN: 978-0-86238-256-8.

[81] Rachael Tatman. *Evaluating Text Output in NLP: BLEU at your own risk*. URL: https://towardsdatascience.com/evaluating-text-output-in-nlp-bleu-at-your-own-risk-e8609665a213 (посећено 21. 8. 2020.).

[82] M. Těšitelová. *Quantitative Linguistics*. Linguistic & literary studies in Eastern Europe. Benjamins Pub., 1992. ISBN: 9789027215468.

[83] *The British component of the International Corpus of English*. Survey of English Usage. 1998. URL: https://www.ucl.ac.uk/english-usage/projects/ice-gb/.

[84] Jörg Tiedemann. „Parallel Data, Tools and Interfaces in OPUS". *LREC*. 2012.

[85] Antonio Toral и др. „Extrinsic Evaluation of Web-Crawlers in Machine Translation: a Case Study on Croatian–English for the Tourism Domain". *Proceedings of the 17th Annual Conference of the European Association for Machine Translation*. Dubrovnik, Croatia, јун 2014., стр. 221–224.

[86] G.A. Troia. *Instruction and Assessment for Struggling Writers: Evidence-Based Practices*. Challenges in Language and Literacy. Guilford Publications, 2011. ISBN: 9781609180300.

[87] A Vaswani и др. „Attention is all you need". (2017.). arXiv: 1706.03762.

[88] D. Vitas. *Prevodioci i interpretatori: (uvod u teoriju i metode kompilacije programskih jezika)*. Matematički fakultet, 2006. ISBN: 9788675890560.

[89] Duško Vitas и др. *The Serbian Language in the Digital Age*. ур. Georg Rehm и Hans Uszkoreit. јануар 2012. ISBN: 978-3-642-30754-6. DOI: 10.1007/978-3-642-30755-3.

[90] R. J. Williams и D. Zipser. „A Learning Algorithm for Continually Running Fully Recurrent Neural Networks". *Neural Computation* 1.2 (1989.), стр. 270–280.

[91] Yonghui Wu и др. „Google's neural machine translation system: Bridging the gap between human and machine translation". (2016.). arXiv: 1609.08144.

[92] K. Yao и др. „Depth-Gated Recurrent Neural Networks". 9 (2015.). arXiv: 1508.03790.

[93] George K. Zipf. „Human behavior and the principle of least effort. Cambridge, (Mass.): Addison-Wesley, 1949, pp. 573". *Journal of Clinical Psychology* 6.3 (1950.), стр. 306–306. DOI: 10.1002/1097-4679(195007)6:3<306::AID-JCLP2270060331>3.0.CO;2-7.

УНИВЕРЗИТЕТ У КРАГУЈЕВЦУ
ПРИРОДНО-МАТЕМАТИЧКИ ФАКУЛТЕТ
ИНСТИТУТ ЗА МАТЕМАТИКУ И ИНФОРМАТИКУ

Завршни рад под називом ПРИМЕНА РЕКУРЕНТНИХ НЕУРОНСКИХ МРЕЖА У ОБРАДИ ПРИРОДНИХ ЈЕЗИКА

одбрањен је _____.

МЕНТОР:

_____
др Татјана Стојановић, доцент, ПМФ Крагујевац

ЧЛАНОВИ КОМИСИЈЕ:

_____
др Вишња Симић, доцент, ПМФ Крагујевац

_____
др Милош Ивановић, ванр. професор, ПМФ Крагујевац

Завршни рад је оцењен оценом _____.