Зборник радова Факултета техничких наука, Нови Сад

UDK: 004.9
DOI: https://doi.org/10.24867/30BE32Zaric

АЛГОРИТАМ ЗА ТРАНСФОРМЕР ЗА ПРОБЛЕМ ПРЕВОЂЕЊА СА ЕНГЛЕСКОГ НА СРПСКИ ЈЕЗИК

IMPLEMENTING TRANSFORMER ALGORITHM FOR THE TRANSLATION PROBLEM FROM ENGLISH TO SERBIAN LANGUAGE

Драган Зарић, *Факултет техничких наука, Нови Сад*

Област − ЕЛЕКТРОТЕХНИКА И РАЧUНАРСТВО

Кратак садржај − *Тема овог рада је имплементирање алгоритма за трансформер за проблем превођења са енглеског на српски језик и формира-ње претренираног модела који се може користити за овај и остале задатке обраде природних језика. Структура која је имплементирана овим радом је предложена у раду [1].*

Кључне речи: *Обрада природних језика, Трансформер, Енкодер-декодер модели, Превођење тек-ста*

Abstract - *This paper topics is the implementation of a transformer algorithm for the problem of translation from English to Serbian language and the formation of a pretrained model that can be used for this and other natural language processing tasks. The structure implemented in this paper was proposed in the paper [1].*

***Keywords*:** *Natural language processing, Transformer, Sequence-to-sequence model, Translation task*

1. УВОД

Обрада природног језика је научна подобласт која користи знање из вештачке интелигенције, информатике и лингвистике. Она се бави способношћу машина да интерпретирата, манипулише и разуме људски језик онако како је написан или изговорен. Данас се за задатке обраде природних језика најче-шће користе претренирани (енг. *pretrained*) модели тј. модели који су претходно обучени над великим скупом података. Затим се такви модели фино подешавају (енг. *fine-tuning*) за специфичан задатак користећи пренесено учење (енг. *transfer learning*). За фино подешавање потребан је мањи број пода-така, самим тим и мање времена за обучавање и мање ресурса се троши. Циљ овог рада је да се кре-ира претрениран модел за превођење са енглеског на српски који касније може бити коришћен за решавање разних проблема обраде природних језика.

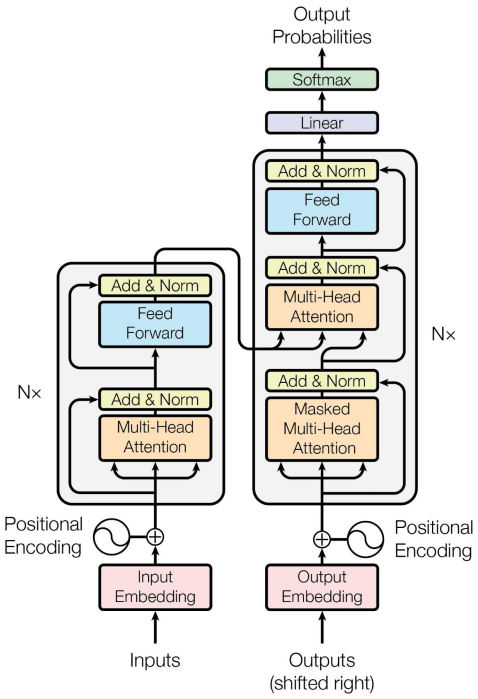
НАПОМЕНА:
Овај рад проистекао је из мастер рада чији је ментор био др Дарко Чапко, ред. проф.

2. СКУП ПОДАТАКА

Први корак у решавању датог проблема је проналажење прикладног скупа података. Трансформе-ри се састоје од великог броја подесивих параме-тара па је за њихово ваљано обучавање потребан велики број подтака којим ће модел бити похрањен приликом обучавања. Након прикупљених сирових података (преузетих са сајта [4]) потребно је кон-вертовати текст у бројеве које модел може да ра-зуме. Тај поступак назива се токенизација. У овом раду је улазни скуп података подељен тако да свака реч чини један токен. Вокабулар који је добијен је садржао 30.000 токена. Поред токена из скупа по-датака додати су токени *[SOS]* за почетак секвенце,*[EOS]* за крај секвенце, *[UNK]* за непознате токене и *[PAD]* токени који не носе никакво значење него служе да свака улазна секвенца буде исте дужине. Осим тога, уз сваку секвенцу приложена је и маска за пажњу (енг. *attention mask*) која се на њу односи. Она говори који део секвенце носи информацију а који не (*[PAD]* токени) [2].

3. ТРАНСФОРМЕР

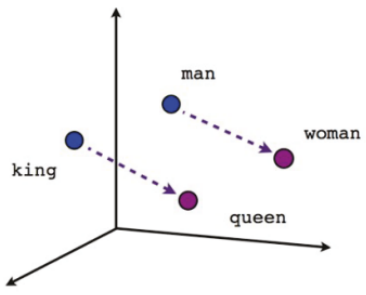
Трансформер је тип модела машинског учења ко-ји се користи за задатке обраде природних језика (енг. *NLP*). Први пут је представљен у раду [1] и од тада се искључиво он користи за решавање про-блема из поменуте области. Ако бисмо посматрали трансформер за проблем превођења, он је као *black box* који на улазу добија реченицу на енглеском језику а на излазу даје адекватан превод те речени-це на српски језик. Унутрашња структура транс-формера се састоји од низа енкодера и декодера. Енкодер служи за претварање улазних података у нумеричку репрезентацију који модел може да ра-зуме а декодер из тих нумеричких репрезентација добија излаз односно преведену реченицу. Унутар сваког енкодера и декодера се налазе механизам па-жње (енг. *attention mechanism*) и неуронска мрежа са пропагацијом сигнала унапред (енг. *feed forward neural network*). Механизам пажње служи да модел схвати контекст и однос између речи из улазне се-квенце а неуронска мрежа за генерално разумевање реченице.



Слика 1: *Структура трансформера [1]*

3.1. Уградња речи (енг. *word embedding*)

Уградња речи је кључна техника за обраду природ-ног језика јер омогућава моделима да претворе дис-кретне језичке ентитете (као што су речи и токени) у континуиране векторске репрезентације које но-се богате информације о значењу и контексту речи. Речи се представљају као вектори у високодимензи-оналном простору. Идеја је да се ухвати семантичко значење речи, тако да речи са сличним значењима буду ближе једна другој у векторском простору. То омогућава да се речи третирају као нумерички вектори, што омогућава извођење математичких опе-рација над њима, на пример њихово поређење преко косинусне сличности или неке друге метрике. Када представимо речи у простору можемо и визуелно тумачити њихово значење и однос неких речи. Са слике 2 можемо закључити да је разлика између ре-чи *"краљ"* и *"краљица"* слична разлици између речи *"мушкарац"* и *"жена"*.



Слика 2: *Визуелно репрепрезентоване речи се могу тумачити [5]*

Уграђивање токена је метода која се користи при-ликом рада са трансформерима: за сваки токен се у табели везује *embedding* вектор који представља ње-гово значење. Димензија *embedding* вектора се на-зива d_{model} и може имати вредности: 256, 512, 768, 1024, 1536, 3072 итд. Већа димензионалност најче-шће значи бољи квалитет енкодовања, али и дуже тренирање. У овом раду је $d_{model} = 512$.

3.1.1. *Positional embedding*

Много о значењу неке речи можемо закључити на основу њеног односа са другим речима у оквиру јед-не реченице. Позициона уградња служи да модел не види улазни вектор као само колекцију токена већ колекцију токена који су у неком односу тј. редосле-ду. Сваком токену у улазној секвенци се придружу-је додатни *positional embedding* вектор који садржи информацију о раздаљини токена са сваким дру-гим токеном. Тај вектор је фиксан, једном се изра-чуна и користи се такав приликом тренира- ња и закључивања и исти је за сваку реченицу. С друге стране, *embedding* вектор се мења приликом трени-рања. Да се не би памтила два вектора, positional embedding и input embedding вектори се сабирају и такви улазе у модел (што значи да морају би-ти истих димензија тј. d_{model}). С обзиром на то да се *positional embedding* вектор не мења он се може посматрати као пристрасност (енг. *bias*). У изради овог рада је коришћена следећа формула [1] за рачунање *positional embedding* вектора:

$$PE_{pos,2i} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE_{pos,2i+1} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right),$$

где је *pos* позиција, а *i* је димензија. Свака ди-мензија позиционог кодирања одговара синусоид-ној функцији. Дакле, коначни улаз у трансфор-мер представља збир *positional embedding* вектора и *input embedding* вектора.

3.2. Механизам пажње (*attention layer*)

3.2.1. *Single-head attention*

Механизам пажње служи за разумевање улазне се-квенце тј. контекст токена у односу на све остале токене и омогућава моделу да се фокусира на ра-зличите делове улазне секвенце док обрађује подат-ке. У случају превођења са једног језика на други битно је схватити које речи су релевантне у извор-ној и преведеној реченици, то ради тако што се фо-кусира на различите токене унутар сваке секвенце додељујући им различите тежине на основу њихо-ве релевантности и важности за тренутни задатак. Пажња се израчунава по следећој формули [1]

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

где су Q, K и V вектори упита, кључа и вредно-сти који се добијају множењем улазног вектора са матрицама упита, кључа и вредности чији се пара-метри добијају током тренинга. *Softmax* функцију користимо јер на излазу даје вредности у опсегу (0,1) док је сума свих вредности 1. Захваљујући то-ме, излаз из механизма пажње можемо посматрати

као вероватноћу да је сваки од токена тачан излаз.

3.2.2. *Multi-head attention*

Уколико би механизам пажње рачунао само је-дан контекст за сваку реч постоји велика шанса да би погрешно интерпретирао реченицу. *Multihead attention* дели *embedding* вектор на одређени број *"глава"* и за сваки део вектора посебно рачуна ме-ханизам пажње и на крају те векторе конкатени-ра. Свака *"глава"* садржи читаву улазну реченицу (скуп токена) али гледа другачији аспект сваког то-кена. *Multihead attention* се рачуна по формули [1]

$$MultiHead(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = Concat(head_1, \ldots, head_\mathrm{h})\mathrm{W}^{\mathrm{O}},$$

где је

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V).$$

3.3. *Feed forward neural netword*

Механизам пажње омогућава моделу да се фокуси-ра на релевантне делове секвенце, али сама пажња не обрађује детаљно све информације које је доби-ла. Након ње долази *FFN* која омогућава транс-формеру да разуме сложене односе између подата-ка. *FFN* примењује нелинеарне функције на сваку позицију у секвенци независно. На тај начин, свакој позицији у секвенци даје богатију и комплекснију репрезентацију, што помаже у бољем разумевању целокупног текста. Састоји од две линеарне транс-формације са *ReLU* активационом функцијом из-међу [1]

$$FFN(\mathrm{x}) = max(0, \mathrm{x}\mathrm{W}_1 + \mathrm{b}_1)\mathrm{W}_2 + \mathrm{b}_2.$$

3.4. *Layer normalization*

Након сваког прорачунавања потребно је вршити нормализацију да се све вредности улазног вектора ограниче на опсег од 0 до 1. Додатно, постоје па-раметри γ и β за скалирање и померање тог опсега који служе да појачавају и смањују вредности ради што бољег предвиђања. Поменути параметери нису фиксни и уче се током обуке. Следећа формула [3] која се примењује на свим векторима

$$LayerNorm(x) = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \cdot \gamma + \beta$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i, \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2,$$

где су: x улазни вектор, μ средња вредност улаза, σ^2 варијанса улаза, ϵ мала константа додата ради нумеричке стабилности, γ и β параметри који се уче. Укупан број подесивих параметара модела је 35 милиона.

3.5. Енкодер

Енкодер на улазу добија вектор који представља збир позиционог и улазног уграђивања, затим ти вектори пролазе кроз слојеве енкодера. Сваки енкодер саджи слој пажње и *Feed Forward* мрежу. Излаз сваког слоја енкодера је улаз у следећи слој енкоде-ра осим последњег. Излаз последњег слоја енкодера је улаз у сваки од декодер слојева и представља век-торе упита и кључа приликом њиховог подешавања тежина. Главна улога енкодера је разумевање улазних се-квенци и њихово претварање у апстрактнију пред-ставу која ће бити прослеђења декодеру или сама генерисати предикцију.

3.6. Декодер

Декодер служи за генерисање излаза. Док енкодер прима текст на одређеном језику (у нашем случају енглеском), декодер прима одговарајучу секвенцу на жељеном језику (у нашем случају српском). Обу-чавање декодера се своди на то да се предвиђа сле-дећи токен у односу на улазни. Механизам пажње током обуке може приступити само речима позици-онираним пре у реченици. Након првог механизма пажње, у декодеру постоји и унакрсни механизам пажње (енг. *cross attention*) где су вектори упита и кључа излаз из енкодера а вектор вредности се добија из механизма пажње самог декодера. При-ликом прослеђивања одређене реченице коју жели-мо да преведемо излаз из енкодера се рачуна само у првој итерацији јер се улазна секвенца није про-менила. Улаз у декодер се мења односно додаје се токен по токен, почевши од *[SOS]*. Декодер пред-виђа следећи токен у свакој итерацији. Када се на улазу нађе последња реч из секвенце поступак је завршен.

3.7. Слој пројекције и закључивање

Слој пројекције (енг. *projection layer*) служи да вектор, који је излаз из декодера, преведе у конкретне излазне токене (слова, подречи или речи). Након тога, вектори пролазе кроз *softmax* функцију која служи да се излаз из слоја пројекције (енг. *logit*) мапира на вероватноће да је неки од вектора сле-дећи који треба да буде изабран.

Након добијених вероватноћа за сваки токен из во-кабулара потребно је одредити који ће токен би-ти генерисан узимајући онај са највећом верова-тноћом или неку другу технику. Изабрани токен се детокенизује у стварну реч или карактер и добија-мо читљив текст на излазу. Излаз из декодера који је прошао кроз слој пројекције се пореди за жеље-ним излазом (*label*) и рачуна се колико је модел до-бро предвидио користећи одређену метрику (нпр. функцију губитка односно *loss funtion*). Затим се пропагирају промене тежина уназад и процес се по-навља све док се не обради читав скуп података. За ажурирање параметара користи се *Adam* алго-ритам. Због променљивог корака учења који се при-лагођава градијенту и корекције бајаса овај алго-ритам је стабилан и брзо конвергира. Уз то, *Adam* треба да памти само два покретна просека по пара-метру што га чини ефикасним за меморију, што је

посебно важно за велике неуронске мреже и велики број података.

4. АЛГОРИТАМ

На слици 4 је приказан псеудокод за обучавање трансформера.

```
begin
    улазни скуп података поделити на серије реченица
    иницијализуј модел насумичним параметрима
    for(i = 1 to број епоха)
        for(i = 1 to број серија реченица)
            израчунај излаз енкодера
            for(i = 1 to крај серије)
                предвиди следећи токен рачунањем излаза декодера
            израчунај губитак
            ажурирај параметаре коришћењем Адамовог алгоритма
        покрени валидацију модела
        сачувај тежине модела
end
```

Слика 3: *Псеудокод тренинг петље*

5. РЕЗУЛТАТИ

На сликама 4-8 су приказани резултати тестирања. Краће секвенце (слика 4) је модел успешно превео. У дужим секвенцама (слике 5-8) модел је грешио лица, падеже а непознате речи није попунио.

```
    SOURCE: A friend called me.
    TARGET: - Prijatelj me pozvao.
 PREDICTED: - Prijatelj me pozvao .
```

Слика 4: *Једноставне секвенце лако преводи*

```
   SOURCE: Hey,you Hey,you won what?
   TARGET: Hey,ti hey,šta si pobijedio?
PREDICTED: Hej , kako si li ti uradio ?
```

Слика 5: *Добар контекст али неадекватан превод*

```
   SOURCE: - Where else can they get in here?
   TARGET: Gde još mogu da uđu ovde?
PREDICTED: Gde bi da bi mogao ovde ?
```

Слика 6: *Добар контекст али неадекватан превод*

```
   SOURCE: This night is a night full of terror
   TARGET: Noæas æemo otpoèeti "noæ terora"!
PREDICTED: Noæas æemo " noæ terora "!
```

Слика 7: *Лош скуп ће резултирати лошим предикцијама*

```
    SOURCE: Listen, I love you, baby.
    TARGET: Slušaj, volim te, dušo.
 PREDICTED: Slušaj , volim ti , dušo .
```

Слика 8: *Погрешна лица и падежи неких речи*

6. ЗАКЉУЧАК

У овом раду је имплементиран алгоритам трансформера предложен у раду [1]. Добијен је претрениран модел за превођење са енглеског на српски језик. Тренирање трансформер модела са 35 мили-она параметара захтева значајну количину времена и ресурса. Како би се постигла задовољавајућа тач-ност потребно је користити неколико стотина хиља-да до милион парова реченица за тренинг. Типич-но, моделима ове величине треба између 10 и 50 епоха како би достигли стабилну тачност, уз напо-мену да је потребно пажљиво пратити валидациони скуп како би се избегло преобучавање. Време тре-нирања зависи од хардверских ресурса, попут гра-фичке картице и броја *CPU* језгара, али у просеку би тренирање једног модела са 35 милиона параме-тара траје око недељу дана на снажнијој *GPU* кон-фигурацији. На *CPU* ресурсима, овај процес траје знатно дуже, од неколико недеља до месеци, у за-висности од расположиве снаге.

С обзиром на то да је овај модел трениран над 10 хиљада реченица 1 дан и трансформер је успео да ухвати значење и контекст речи можемо закључити да је ово моћна структура која се може користити за разне проблеме обраде природних језика. Надоградња овог рада ће бити *„чишћење”* скупа по-датака и тренинрање модела над милион реченица.

Литература

[1] A. Vaswani, N. Shazeer and others, *Attention is All You Need*, Advances in Neural Information Processing Systems, 2017, pp. 5998-6008. https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper. pdf

[2] Daniel Jurafsky and James H. Martin, *Speech and Language Processing*, An Introduction to Natural Language Processing, 3rd edition, 2024 https://web.stanford.edu/~jurafsky/slp3/

[3] Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton, *Layer Normalization*, 2016 https://arxiv.org/abs/1607.06450

[4] https://data.statmt.org/opus-100-corpus/v1.0/supervised/en-sr/

[5] Слика преузета са: https://www.brainlabsdigital.com/blog/what-word2vec-means-for-seo/

Кратка биографија:

Драган Зарић рођен је у Ужицу 2000. године. Мастер рад на Факулте-ту техничких наука из области Елек-тротехнике и рачунарства одбранио је 2024. године.
Контакт: zaricdragan00@gmail.com