1. Увод

Циљ овог пројекта је давање смислених одговора на корисникове поруке у реалном времену. Корисник не треба да буде свестан да разговара са *chatbot*-ом, већ са правом особом. Како је као *dataset* искоришћен скуп података *Customer Service Support*(о овоме детаљније у наставку), *chatbot* одговара на поруке везане за корисничку подршку највећих брендова, конкретно *Apple*-а , *Amazon*-а и *Uber*-а.

Мотивација за овај пројекат је била то што је данас *chat* најучесталији вид комуникације. Овај вид комуникације је динамичнији од нпр. *email* комуникације. Такође, корисници ће пре изабрати *chat* као начин да дођу до додатних информација.

2. Методологије

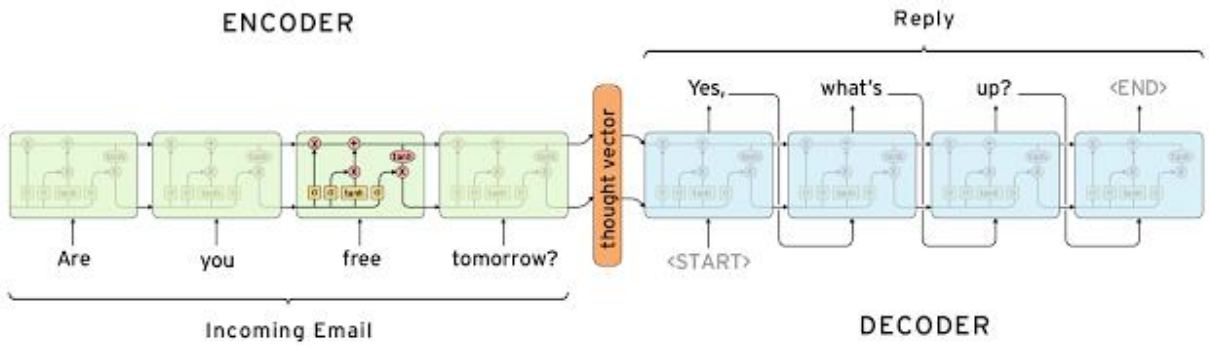
Решавање проблема врши се кроз два дела:

1. Претпроцесирање података
2. Креирање кодер-декодер архитектуре

Главни део решавања овог проблема било је креирање кодер-декодер архитектуре. Ову архитектуру смо дизајнирали тако да користи рекурентне неуронске мреже са *LSTM* ћелијама.

2.1 Енкодер и декодер

Енкодер преузима реченицу као улаз и процесира по једну реч у јединици времена. Идеја јесте да се секвенца речи (симбола) конвертује у *feature* вектор фиксне дужине који енкодира само битне информације у секвенци док губи непотребне информације. Сваки *hidden state* утиче на следећи. Последњи *hidden state* се може посматрати као сумаризација читаве секвенце. Назива се и контекст. На основу контекста, декодер генерише нову секвенцу, један симбол (реч) по јединици времена. У свакој јединици времена, декодер генерише нови симбол на основу контекста и на основу претходно генерисаних симбола.



Слика 1. Енкодер-декодер архитектура

2.2 Вокабулар

Битна ствар у оквиру *seq2seq* модела јесте вокабулар. Вокабулар се састоји из различитих речи и симбола присутних у *dataset*-у.

Како изабрати величину вокабулара? Свака реч која се не појављује у вокабулара биће замењена са *unk* токеном (*unknown*). Број ових токена у *dataset*-u прави велике разлике. Ако постоји велики број непознатих токена, модел ће научити да у резултат убацује непознате токене више него речи. Прихватљиво је да број ових токена буде испод 5%.

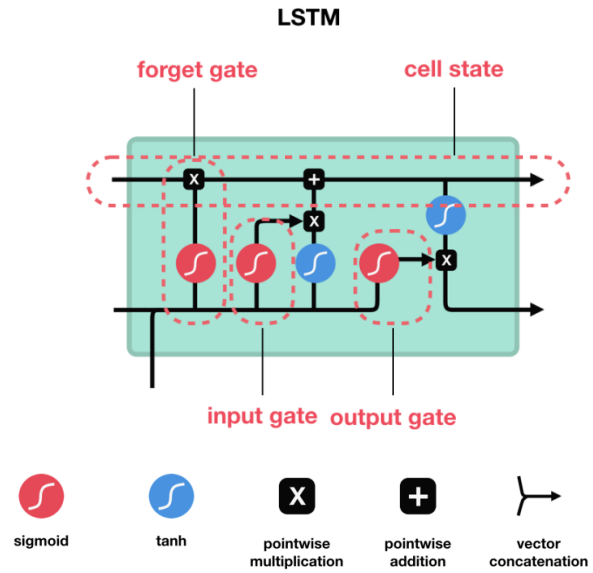
У нашем пројекту, број различитих речи на тренинг скупу података износио је 20000+ симбола, односно речи за тренирање *Apple*-а и *Amazon*-а. Величина вокабулара у пројекту за коју смо се одлучили је 20000, како би било што мање непознатих токена.Са друге стране, број различитих речи код података у вези са *Uber*-ом је износио око 10.5 хиљада. Величина вокабулара за коју смо се одлучили у овом случају је 10 хиљада.Речи које нису убачене у вокабулар су биле речи које су се ретко појављивале, тј. чија фреквенција појављивања у *dataset*-у је била мала.

2.3 Long short term memory networks (LSTM)

LSTM је посебна врста *RNN*-а. На слици 2 је приказан изглед једне овакве ћелије. Битан део *LSTM* је стање ћелије и гејтови. Стање ћелије можемо назвати меморијом мреже. У теорији, стање ћелије може да носи релевантне информације услед процесирања секвенце симбола. На овај начин информације из ранијих јединица времена, могу да се нађу у каснијим јединицама времена и на овај начин се редукују ефекти краткотрајне меморије. Услед процесирања секвенце, стању ћелије се додају или уклањају информације помоћу *gate*-а. Капије одређују које информације могу да се налазе у стању ћелије. Како су и *gate* неуронске мреже, оне могу да науче које информације су потребне да се задрже, а које да се одбаце у току тренинга.

Gate укључује сигмоид активацију. Сваки податак помножен са нулом ће дати нулу, што резултује заборављањем. Сваки податак помножен са јединицом ће дати сам тај податак што резултује памћењем. Постоје три врсте *gate*-а:

1. **forget gate** – одлучује које информације ће бити одбачене, а које задржане
2. **input gate** – за ажурирање стања ћелије
3. **output gate** – одлучује који је следећи *hidden state*

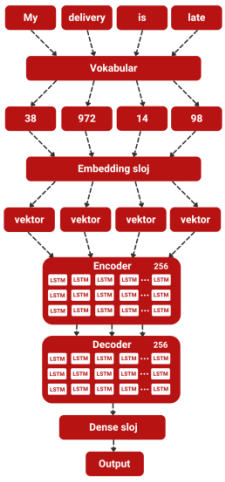


Слика 2. *LSTM* ћелија и њене операције

2.4 Наша архитектура

За решавање овог проблема користили смо енкодер-декодер архитектуру. Енкодер и декодер се састоје од по једног *RNN*-а. Мрежу смо обучавали кроз 40 епоха. Користили смо *Adam optimizer* са *learning rate*-ом 0.001. Величина вокабулара износи 20000.

Речи у реченици су представљене бројем који се користи у моделу приликом обраде реченице. Речи заједно са бројевима који их одређују су смештени у вокабулар. Вокабулар је креиран помоћу *keras* библиотеке. За речи је одређено 1024 речи које се користе заједно са датом речју у реченици (контекст). Обе мреже имају по три слоја од 256 *LSTM* ћелија. Излаз декодера је улаз у dense слој, где имамо различит број ћелија у зависности од dataset-а који је повезан са 20000 ћелија. Овај слој формира *output*. Мреже су имплементиране помоћу *TensorFlow* и *TensorLayer* библиотека. Слика архитектуре наше мреже може се погледати на слици 3.



Слика 3. Архитектура мреже

3. Скуп података

Скуп података који је коришћен у пројекту је *Customer Service Support.* Овај скуп података броји преко 3 милиона твитова и одговара на исте у вези са корисничком подршком највећих брендова. У скупу података се налазе подаци за различите брендове попут *Apple*-а, *Amazon*-а, *Uber*-а, *Delte*, *SpotifyCares*-а итд. Податке које смо одабрали за тренирање наше мреже јесу подаци који се односе на корисничке подршке *Apple*-а, *Amazon*-а и *Uber*-а. Података за претходна три наведена бренда има највише, па смо се зато одлучили баш за њих.

3.1 Претпроцесирање података

У оквиру припреме података за тренирање неуронске мрежу, урадили смо следеће кораке:

1. **Пребацивање речи у lowercase** – како се не би различито биле третиране исте речи написане у *lowercase*-у и у *uppercase*-у.
2. **Замена сленга и скраћених облика речи** (попут *faq*, *g9*, *gn*, *don't*…) – како би мрежа исто третирала нпр. *gn* и *good night*
3. **Замена урлова речју url** – како би се исто третирао сваки урл
4. **Уклањање помињања корисника тзв. mention-а** – нерелевантно за обучавање мреже
5. **Уклањање хештагова** – такође смо их сматрали нерелевантнима
6. **Уклањање емоџија и емотикона**
7. **Уклањање бројева** – нису релевантни да се нађу у вокабулару
8. **Уклањање whitespace-а**
9. **Уклањање речи које не припадају енглеском језику** и како би мрежа лакше учила, неисправно написане речи су замењене коректно написаним
10. **Избацивање реченица које су дуже од 20 речи за Amazon податке, 21 за Uber и 25 за Apple**

У плану је било да се уклоне и тзв. *stop words*, као и лематизација. Међутим, у овом случају наш *chatbot* није давао довољно смислене одговоре.

Даље, варијабилне дужине реченица су пребачене у реченице исте дужине помоћу *padding*-а. Користили смо следеће симболе да попунимо реченице:

- **start** – почетак реченице
- **end** – крај реченице
- **unk** – непозната реч, не постоји у вокабулару
- **pad** – филер

Иако смо свели реченице на исте дужине, јавио се следећи проблем. Уколико је најдужа реченица у скупу података дужине 100, онда ће све реченице, чак и оне које се састоје од свега неколико речи требати да се представе као реченице од 100(ово се добија додавањем симбола *pad*), међутим, у овом случају ће бити прикривена информација коју носи та кратка

реченица. Стога смо се одлучили на коришћење *bucketing*-а, који решава овај проблем тако што ставља реченице у бакете.

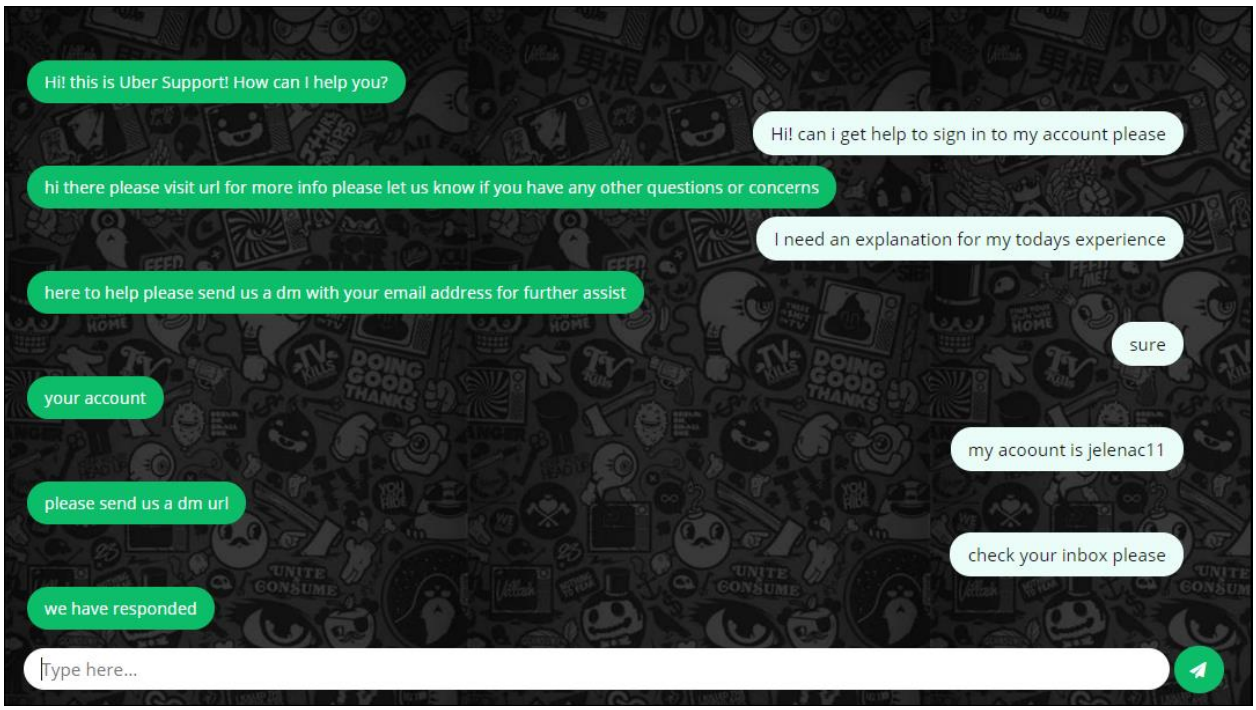
Такође, коришћењем ембединг слоја за сваку прослеђену реч одређује се и колико су њени суседи удаљени од ње.

Библиотеке које су коришћене у овом делу претпроцесирања података су *nltk, spellchecker, pyspellchecker*, *emot* и *keras*.

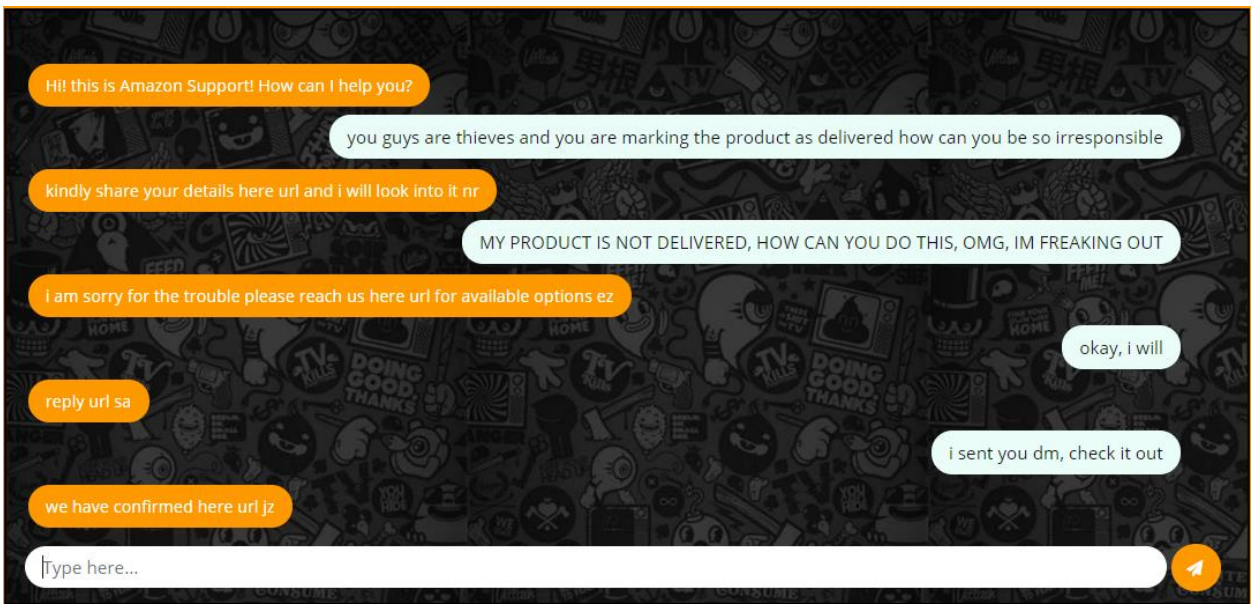
Скуп података је подељен на 90:10, где је 90% података искориштено за тренинг, а 10% података за тест.

4. Резултати

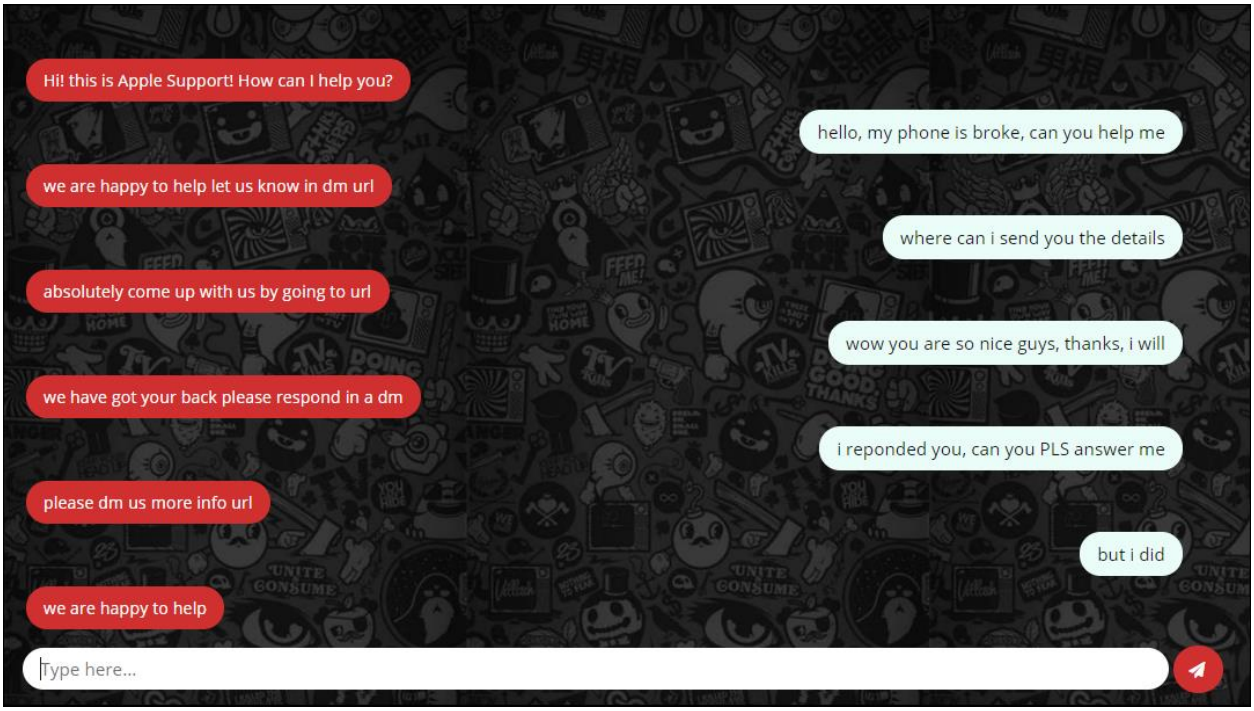
Како на иста питања може бити одговорено на различите начине, за валидацију је коришћена самоставлна валидација и *loss*. *Loss* који смо добили на крају је 0.3 за *Uber,* 0.5 за *Apple* и 0.6 за *Amazon*. На следећим сликама су приказани неки од резултата.



Слика 4. Резултати *Uber*-а



Слика 5. Резултати *Amazon*-а



Слика 6. Резултати *Apple*-а

*Кориснички интерфејс је развијен помоћу *Flask*-а.

5. Закључак

Мрежа углавном смислено одговара на постављена питања везана за корисничку подршку. Проблем који постоји јесте што дуге конверзације нису могуће и што мрежа није способна да одговара на питања која нису уско повезана са корисничком подршком.

Даља унапређења овог пројекта се могу реализовати обучавањем мреже и за давање одговара везаних за друге корисничку подршку других брендова. Такође, могуће је још оптимизовати коришћене хиперпараметре и користити неки *attention* механизам као што је нпр. *Luong Style Attention Mechanism*.

6. Референце

- http://suriyadeepan.github.io/2016-06-28-easy-seq2seq/
- https://medium.com/swlh/how-to-design-seq2seq-chatbot-using-keras-framework-ae86d950e91d
- http://suriyadeepan.github.io/2016-06-28-easy-seq2seq/
- http://suriyadeepan.github.io/2016-12-31-practical-seq2seq/