Poetry Generation on Kazakh Language

1st Aben Sadykhanov
Department of Computational and Data Sciences
Astana IT University
Astana, Kazakhstan
211272@astanait.edu.kz

2nd Arman Nurken
Department of Computational and Data Sciences
Astana IT University
Astana, Kazakhstan
211282@astanait.edu.kz

Abstract—**Firstly in the world there is created poetry generator on Kazakh language. It is used character-based Recurrent Neural Network (RNN) with Long-Short term memory (LSTM), and Dropout layers. Training data-set consists of poems of dozens of Kazakh classic poets with 450 thousands characters. There is generated real poetry, not like other generators where no rhyme and rhythm. It has rhythm, rhymes and rhyming structure which specific for only Kazakh poets "Қара ұйқас". Generation has few grammatical errors, meaningful word phrases and abstract meaning.**

Index Terms—**Poetry generation, Recurrent Neural Network, Long-Short term memory, poetry generator, Kazakh Language**

I. INTRODUCTION

Knowledge of the Kazakh language is used rarely, let alone globally, but there are also problematic situations in Kazakh society that need to be addressed. To partially solve this problem, there is poetry, which usually uses rich and creative language, introducing new words, metaphors and descriptive phrases. Thus, it plays a vital role in language development by expanding vocabulary, developing creativity, exploring sounds and rhythm, conveying emotions and meanings, protecting cultural heritage and challenging the use of traditional languages.

Since there is not much research and projects related to the Kazakh language almost globally, such as immersive reading, automatic spelling corrector, text generation, and poetry generation, we decided to inspire our generation with a generative poetry program using NLP (natural language processing), including its the most intuitive way, RNN (recurrent neural networks) and LSTM (long-short term memory). And also, we test other architectures in order to compare our main architecture and make sure that it works better, exactly, on Kazakh language, such as: GRU (gated recurrent unit), Bidirectional LSTM. We are happy to develop the use of the Kazakh language not in our familiar society, but on a global scale. We remember with poetry in Kazakh society such exciting poets as Abay Kunanbayev, Sultanmakhmut Torayghyrov, Mukagali Makatayev, Ilyas Zhansugurov etc. So why do not we study their poems and use them as the main data set, and train the models we built by their structures of created poems (rhymes, control meter, semantics) into neural networks.

As we mentioned on the idea description, we decided to work in field of NLP (natural language processing) project, which generates human-like poetic texts in Kazakh. The goal is to create poetry that are indistinguishable from human-written poems. We use RNN's (recurrent neural networks) two methods: word-based and character-based. And LSTM (Long-short term memory) layers to eliminate the problem of long-term addiction.

Good example from Karinka Kapoor [1], lyrics generator based on LSTM with character-based model. There is good explanation how to deal with data preprocessing, that for the model it is reliable to give sequence of the string, data in one column. Removing not required symbols, so that she mentioned that data cleaning process for NLP is 'crucial' preprocessing. Once the counting of unique characters is done, then by looking to the collection, we should consider what we should remove, so that foreign language symbols and irrelevant ones that we do not need, we should to remove. The next thing that computer does not understand text, so that it is just number of clusters. For this we should create mapping dictionaries with consisting of encoded unique symbols. She came to conclusion that model does learn to structure of the poem, but most of the results was meaningless. Despite of this character-based model generates some legitimate words.

Andrej Karpathy [2] has an interesting article for poem generation, who is training RNN all the time, almost 1 year. Which the work based on multi-layer LSTM and character-based model and GRU. He mention that the sequence regime of operation is much powerful and hard than we think, compared to fixed networks that uses fixed computational steps. On the data prepossessing moment we extremely get our point back, so that the model should and must learn any English or Kazakh language from scratch, including where to put commas, apostrophes, spaces and etc. There is parameter that is called temperature (and we explain about it in the methods section), decreasing its number, closer to 0.5, gives chill to RNN, that makes more confident, but also predictable result. The higher temperatures will give more diversity, but with cost of more incorrect predictions. In overall LSTM can spell the words and learn its syntactic structure.

By several recommendations we also consider transformers-based RNN, which uses GPT-2 (generative pre-trained transformer) models. A good example provides Koziev [3], on poem generation in Russian language. There was several ruGPT and ruT5, that generated worst and did not learn for phonetics. In development such models can remember and generate mostly only pair rhyming words, but it is not enough for poems:

Я знаю, ты не станешь смеяться

В этот день над моими стихами.
Только белая берёза проснётся
Под моим окном, и заплачет с нами

GPT-2 model gets acquainted with phonetics of Russian language, by dividing words to syllables and rules of alternation of stressed and unstressed syllables. In overall, Words divides into syllables, and syllables are unfold from right to left for each line of the poem:

Я оставляю брошенные фразы
Иного смеха, слабости и слёз
Я превращаюсь в голубые стразы
Кружась ветвями молодых берёз

The rest of the paper focuses on gathering data and finding sources for it. Certain pieces of poetry in the Kazakh language will be highlighted. The generation of poetry using techniques and architectures like RNN, GRU, and LSTM will then be covered. Using these techniques, models are built and trained on data-sets, and the resulting poetry is examined for rhythm, rhyme, and meaning by survey among university students.

II. DATA-SET AND SPECIFICITY OF KAZAKH LANGUAGE

To train neural networks, there are always used big data-sets. Actually, on Kazakh language data-sets are rare thing, it means that we should create own data-sets. We only consider poems because they are large and consist of stable syllables. We tried to collect poems which have only 10-12 syllables. Conducting experiments and playing with collecting poems with different types of syllables in a line, we saw unstable results (we also show it in the results section), and those that we do not need to meet. Moreover, We use poems which written only by using "Kara Uykas". It is specific type of rhyming which used often in kazakh poetry. "Kara Uykas" have rhyming structure AABA. It means first. second and forth lines are rhymes, but third is not rhyme.

Data-sets will consist of the poems of Kazakh Classical poets. For instance, Abay Kunanbayev: "Masgut", "Azim"; poems and songs of Mukagali Makatayev: "Ilyich", "Aqqular Uiqtaganda", "Bolshevikter", "Altay-Atyrau", "Rayimbek! Rayimbek!"; Sultanmakhmut Toraygirov: "Tanystyru", "Adasqan omir", "Kedey", "Aitys", "Qamar Sulu", "Kim zhazyqty"; Magzhan Zhumabaev: "Ertegi", "Oqzhetpestin qiyasynday", "Batyr Bayian", "Qolybaydin Qobyzy"; Mukhtar Shakhanov: "Tanakoz", "Kure tamyrdy izdeu"; Saken Seyfullin: "Kokshetau"; Ilyas Zhansugirov: "Kuyshi", "Qulager"; Sabyt Mukhanov: "Sulushash"; Mukhamedzhan Seralyn: "Gulkashyma". If you are familiar with the works of Kazakh poets or curious of collecting poems, we did not forget about main figures, named "Bes arys", "Ush bayterek". And one thing that we considered only poems and there appear several famous authors with no poems, which are: Ybyrai Altynsaryn, Alikhan Bokeikhanov, Ilyas Esenberlyn, Olzhas Suleymenov, Akhmet Baytursynov, Myrzhaqip Dulatuly, Sapargaly Begalyn. Poems will be collected from sites bilim-all.kz, zharar.kz.

III. METHODS AND TECH STACK

A. Data collection and cleaning

There were collected 25 poems, written by 10 poets, which is more than 450 000 characters. Were collected poems with only 10 - 12 syllables per line. In the cleaning chronological dates, headings, lines that breaks syllable limit in some poems were removed. Punctuation, specific symbols were deleted, that was left only 42 symbols (letters). All data text were converted to lower case, to prevent any incorrect generating and underfitting on the training part. By the principle of character - level model of RNN, every character must be converted into 'integers', so for this there 2 dictionaries to which will be saved "char2int", "int2char" vocabularies. It means that each unique character will be encoded to "int". We perform Exploratory Data Analysis, to get to know better about our data-set, as we work with text data it is better to plot word cloud:

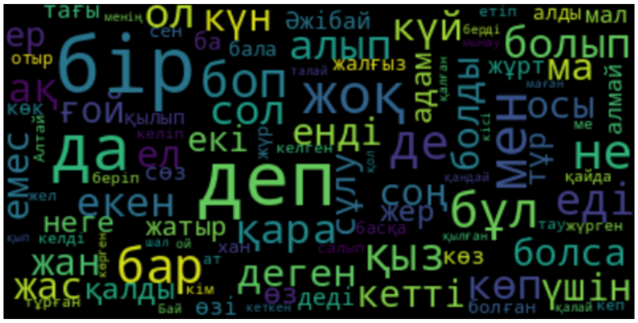

Fig. 1. Exploratory Data Analysis

B. RNN

Recurrent neural network [4] has one vector for the item which also is output and deal with sequential data to make predictions. Also it is able to memorize some parts of sequence, to make accurate predictions. It popularly uses on text, poem generation, speech recognition, translation and etc. So, how it works? [5] It learns from example on the data-set, takes it and apply some computations, like using random taken variables (weights and biases). After then, predict the next word, then the next word, then the next word... Which means the prediction depends on previously seen elements.

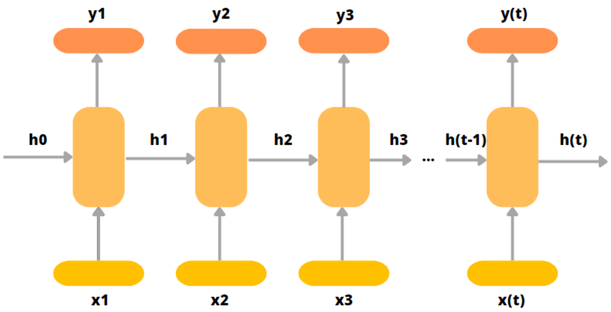

Fig. 2. RNN architecture

Here X1, X2, X3 represent input data from the text, Y1, Y2, Y3 represent predicted next words, and h1, h2, h3 holds information to the previously taken input words. [6]

C. LSTM

With a large amount of data, RNN becomes unusable, since it remembers the last or next received information rather than the previous or long-past one. Therefore, LSTM comes to replace this neural network. Long Short-Term Memory unit is made of a cell that consists of an input, output, forget gate. With applying the same weight to each part (chunk), hidden state storing the prior text that is updated, adding new information(text), removing irrelevant data. It is responsible for memorizing only certain time of period and regulate how much data is kept. There is two main approaches in training. First character by character, which gets input as a sequence of characters and by looking the first or previous character tries to predict the next one. Second level is word by word, which is actually the same approach as the character-based level, except for prediction, which is related to predicting the next word.

Here is how it works in architecture step by step:

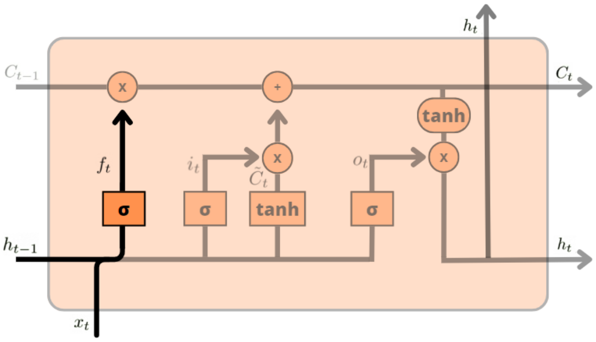

Fig. 3. Forget gate

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

Forget gate looks at h(t-1) and x1, then returns values 0 - 1 for C(t-1).

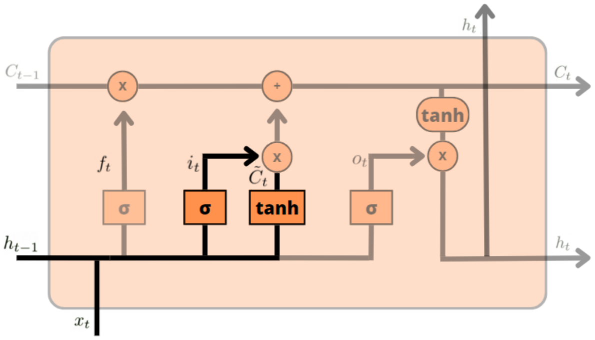

Fig. 4. Input gate and candidate for cell state update

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{2}$$

$$C'_t = tanh(W_C * [h_{t-1}, x_t] + b_C) \tag{3}$$

Sigmoidal layer determines which values should be updated, Then the tanh layer builds a vector of new candidate values C(t).

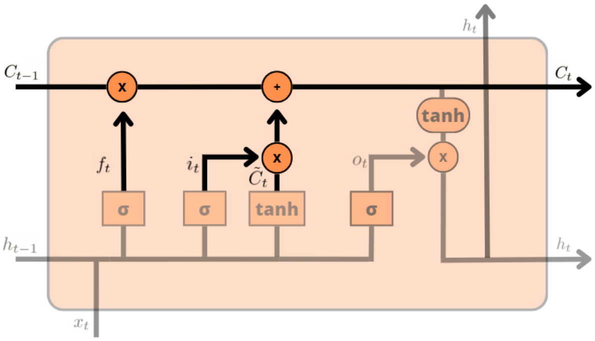

Fig. 5. Long-term memory

$$C_t = f_t * C_{t-1} + i_t * C'_t \tag{4}$$

Replace the old C(t-1) to C(t) multiply the old f(t), to forget the unnecessary. Then add i(t)*C(t), the new candidate values.

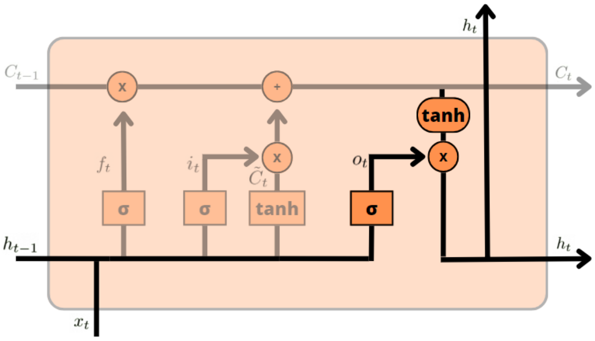

Fig. 6. Output gate

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

Sigmoidal layer selects necessary info. Tanh layer to get values from -1 to 1 at the output, and are multiplied with the output values of the sigmoid layer.

There is two ways to split the data. First is poem by poem, that each poem will be considered and will train separately and length of each poem will be limited equally, so for the training it would be reliable, but more complex and takes a lot of time, computational power. And the second is to giving the sequence length by seed (giving example with which poem will be generated), by which model will take splitting. One of the negative thing that occurs is that model will not look at the data-set as a whole poem, therefore it meets in some lines lose of the poem context.

We can take experiment-analyse as a basis of our project provided by computer engineering students at "Istanbul Bilgi University". Participants were fully aware of the purpose of the test:

"An experiment is made with 146 participants to test if our automatic poetry generation program ROMTU is able to achieve this goal. As a result, ROMTU were able to mislead 48.63% of participants.For creating a lexicon, firstly 1500 different poems are gathered from siirakademisi.com website." [7]

Participants ranked each poem from 0 to 5 (0: weak, 5: strong) according to following criteria: Rhyme, Message, Usage of Language. As a result the survey gives us out of five 3.28 for the rhyming, 3.25 points for the message and 3.30 for the language usage.

To get a good result, we will take the generated poems and human-made poems and compare them with the created groups of professionals and ordinary people, the assessment will represent the probability of a true poem.

Kiis, Kängsepp [8] created a poem generating program using different neural networks. For the use of LSTM 2 main approaching models, that was mentioned, in the word-based model consisting 2 layers and 500 hidden units on layer with 0.2 dropout gives the best results with less random and broken words, but the result was not very good, where it almost did not learn very well haikus (short poem) structure and almost did not make sense. It did not output an ending verb, that just takes 3 beginning columns of the rows:

Poem (1000, 2 layers, 30 seq, 0.2 drop)
raped and terrified,
because the blood of the young men school,
and mangled language, crying, full of pain,
and maimed begin it to be a fool.
the future seems so strong full of life.
the world take swimming out of the world
to take care of our own life.
the one that's right is getting removed from colorful.
that said, 'oh, what's go and trust?'
he loves to count

Kiis, Kängsepp [8] tried also character-based LSTM with sequence length from 100 to 400, with dropouts and additional layers. In comparison, for the result it gives good ones, almost correct punctuation, suitable poem structure, but some words were predicted wrongly due to random nature:

Poem (500, 2 layers, 100 seq, 0.1 drop)
Amid seed animals and hearts to live,
In this curious paradise of light fame
God dares to be simply purified
Without buried tears like solemn brief
On human God, the killer's Jap

As for the remark the large sequence length will increase randomly choosing, which led to context and structure loosing, and incorrect predicting for words. But if the length of the sequence is small, then there is a lack of an example to generate. Overall both character, word-based models were varying a little, where in character-based we have better poem structure, word predicting. But the most important moment was on the meaning that was lacking. And it is hard to say which parameters gives the best result, whereas all results were similar.

D. GRU

Gated Recurrent Unit is a type of LSTM, that uses 2 gates, which is reset and update, while LSTM uses 3 gates (input, output, forget). It takes less learning parameters, less computational resources, does work and learns faster than LSTM. If amount of the data-set is large and accuracy is important, then usually LSTM in use. But as an alternative GRU can handle a part of the large data, for the use of less memory and faster work, and can regenerate result until it gets better evaluation.

Comparing thing that on GRU we used temperature level, which evaluated labels from 0 to 1. Based on temperature level model predicts to reach end symbol or length limit. If the temperature is close to 1 the output is likely randomly chosen, whereas if it is closer to 0 then the output is likely already spoken how to predict. In addition, if we decrease temperature value it becomes predictable, and so often generates an existing poem, remembers placing index of the word, consisting characters. For example if the word so often places in the third column by count of the words, and if those characters uses in that word, it gives the similar result as it appears in the data-set. On the other hand if we increase the temperature's value, the prediction will become more 'creative', but there is the the risk of having nonsense result.

Piero Paialunga [9] (musician and data scientist) made song lyrics generator on GRU with Embedding, such that words are seen in vectors and should be put in the best way, Dense layer with the logit that gives most probable word that we need to expect.

As we see the result with temperature 0.8, it generates more surprising poem, correct predictions, learns structure from the data-set, seems that context is a little bit exists, but crying thing that rhyming appears rare, that the program mixes whole ending words, verbs, and prediction to the correct rhyme will be too large. Here idea comes, that we should try to collect data-set with same rhyming structure, where others could not teach to rhyming.

Why?
In the garden, would you trust me, me
Red lips and rosy cheek
Say a mind of my friends are saying
Girl, what are you thinking?
You're better off
You're better off, you're be the man
I'd be the man
Don'beat wet you clean
I still got you all over me
Drop everything now
Meet me

IV. RESULTS AND DISCUSSION

In this section will be provided many architecture and models which was trained on different data-sets. I should notice that all models trained the same epoch number. We will provide some generated examples with analysis and this examples are best from all samples we created. All examples are on Cyrillic alphabet of Kazakh language.

A. GRU

We trained RNN with GRU layers on full data-set (450000 characters). There are some examples of generated poetry.

Example 1:

ала алмай надандықпен есіме емес бұл қалайда
ақ көбік салған еді сол мұңды айдап
кетейін беріп қалсақ жылар жалғыз
қарағайдың қаныш кезек астына құлап тұрған
үйінен екі көзі ақ суынан шыққан кісі
қызықсам күйді тартты құлап түсті
алты алаштың жүрегінде дерт екен
ертерек талпындаған қара жер

Example 2:

алашта тұлайым деп қалың қалмақ
кетеді екен алтай қатыныңа
бұл күнде бүтін емес көңіл шатқа
ағайын бүгін мініп сұм қарғалай
бала жатыр төсекте күйіпжанып
балалық бәрі ойым өрлеп байлап тастан
қалмақтың қайсар қызы қайырылмаған
болатын бір өзіңе бар көреді
жалғызақ кенекем ғой қайда қайтар

There are no grammatical errors. We can see appropriate phrases. But poetry has no meaning and only 2 rhymes: қалайда-айдап; other lines have no rhymes. Moreover it has no rhythm, every line has distinct number of syllables.

B. Bidirectional RNN

Another architecture is RNN with one bidirectional LSTM layer, one dropout layer and one LSTM layer which trained on full data-set. There are some examples of generated poetry.

Example 1:

көп сөз ғана қалай деген жан
көргенде жан күйдірген барады аулап
жастық деп әділдікті ертеді не
арқаның арты менің басы деп есі
бір түрлі оқу жетіп өлгі балақа
тумаймын мені сойлау да
алдарын төмен барып қолым па еді
өлімнен күлтай айтқан сорлы бар ғой жапан
мал менен мүмкіт едім қазір деген
кісіңді сізге тоқса бермес еді
сөйлескен сәуле төнді
бір жаса сонау болса кетті малды
жұрт етіп әлде өзіңе

Example 2:

бір кісі ойлай бер деп сен
тұрмаған оған барып қаптасады
тым болмаса сол жастың орып ал
бізге де алдап басып көріп
сөйтіп бір түзел ала қашырады
бір жортқан болар деп асқа
жер қыз боп қара қазақ шаба алмас
бір істі жаз келмеген оны қыстай
апырай ақыбайда баян қойшы
сол жолда қабыш бірін ауып
тамағыңда бізді ғой қарттарына
бір іске енді әңгімен көрінген

Bidirectional RNN works worse than GRU. There are some grammatical errors like "өлгі", "балақа", "мүмкіт", "қабыш". We can see some weird phrases. Poetry has no meaning and no rhymes. Moreover, it has no rhythm, every line has distinct number of syllables. There are lines with 7 and 12 syllables.

C. Small RNN model with LSTM

After training GRU and Bidirectional RNN we try to train RNN with 2 LSTM layers and 1 Dropout on small data-set (actually, this type of architecture would be the most successful, we show it below). It is used Sultanmakhmut Toraygirov's poem "Кедей" which has only 25 thousands characters. There are some examples of generated poetry.

Example 1:

самды арып қайралып ұяқ
кетеді жасалармың қалай
ұмытсы болмаса да сатқандары
тағы аданда тұрайын сорлы адамдай
көтерін беріп түреге керек есін
айлап қарт жүректер тармайды дастар
жем қылай көрсемен терекеген
тұра жүрік кетерік шарар екен
бұл лансыз көк жүзі мендігенінде
дөрлеуден бұл жылқаға бірі болса
тек ұртық екі үйілгі жандарлар
қаралады қоямылған ұртағаны

Example 2:

бұл іске түсірмесін жылым жанып
әкесі алыс емес қажын алар
сандақтас болар мекен байлар жауын
бар болған соң қайта кеп деп қапырған
бұл күнде жақындады жан жамалып
әкесінің бір шығып жерді жанған
екі жетім байырға көрген жанның
көнбесем мынау іздеп ауыр тамақ
жасында жан жамылып әкесі деп
жүргені байлор қосқа заман тұрып
үмітпен параз бермес жаман кетсе
келмедім не боп кетті қыздар едім

The main advantage of this model is that it outputs line with 10-12 syllables. When we use LSTM layers, model understand rhythm. There are some grammatical errors. We can see appropriate phrases, but whole poetry has no meaning. There are more rhymes: терекеген-екен; жанып-жауын, қапырған-жанған.

D. 3 LSTM layers

After seeing successful results, we try to outperform previous results by adding more LSTM layers. Architecture is LSTM-Dropout-LSTM-Dropout-LSTM. This model trained on full data-set. There are some examples. Example 1:

алты алашты алдауға құлақ сағыныстар
көргені сонда жаудан қара жалған
бай болса болса сондай келмес
адамның адам бар ма жан топырап
жанына жау тапқаным бар ма лажыста шоқты
мен сені есін тағы от тамам жоқ
қара жерде табылса кетті қамар
алдымен қара түнде қайта шулап
топырағы мал мен жастық көрінем жасырар
сонда ғана жан емес жан торға ала ма
болса да оны отпасын жасыр байлар
сонда ғана жақсының көшпе тұрмын жауып

Example 2:

байлар ма екенін бас
тарта алса бір кедей кедей жайрап тұрмын
қара жерде табысы көшпелі елден алып
кедей мен сен де көрген көзім жасым
жастықта байлармысың көрге сойға барады жер
мен жастық табылмайды басқа жаққа байлап
сонда ғана жан емес жан торға ала ма
екі жаз мен жас жағы сен туған
жау жақта бар ондай да қара жалған
байларға өткен жарыс алғыс алмақ
жастық жоқ жан жыртылыны жара
таусылар мал кедейлер ақыл беріп

Generally, this model performance is worse than it is considered. There are some grammatical errors like "лажыста". Examples have no rhymes and no rhythm. Some lines has 16 syllables and some 11 syllables. Model do not understand feature rhythm, despite that all data-set has only lines with 10-12 syllables.

E. Final model

After seeing bad results of model 3 LSTM layers, we trained model with initial architecture. Architecture is LSTM-Dropout-LSTM. We understand one thing. Not always bigger model with more layers perform better than smaller model. This model trained on full data-set with 450000 characters. We provided more examples, because it is final model with good results.
Example 1:

көркем күні мынау іші жан тамам
баян бетіп жерге ауыз бар малар
амалсыз адаларға болмас па еді
әжім емес сен тыңдап тере жанған
балаға да болмаса емес мен ма
екі жатыр жаным кеп айтты үйіне
мен емес пе мен істеп пе егіз бар ма
әрине абылайдан құтқанында
екен деген жасым бар таласпады
ой мен күні көресің бұл табар шалға
әлімнен тұмандыққа көзі кемес
бір кезде ақыл тауып аузын тіріп

Example 2:

аққу аққу беті қайғы бар алатын
бұл күнгі бақытсыздық пашын жаныл
біреуміз өмірімен жетер деген
көрсетіп таудан беріп қарағанды
жүрегімен құдай менен тапты беріп
бұл күнде өлең бір жоқ әжібайда
өзіне жас пайдаға бармайсың деп
қызық қып үйге апарып жаратылған
неге сондай көргені жеткен күні
қалып ұман тастады өзі бар ғой
асанның тойғаным да болмас едім
бай қылар деген сөзі алдым шатып

Example 3:

бұл іске түсірмесін жылым жаныл
әкесі алыс емес қажын алар
сандақтас болар ма екен байлар жауын
бар болған соң қайта кеп деп қапырған
бұл күнде жақындады жан жамалып
әкесінің бір шығып жерді жанған
екі жетім байырға көрген жанның
көнбесем мынау іздеп ауыр тамақ

Example 4:

барса салып қылышпан қара күшті
әкесі сүйген сайын өлген күні
үні болса сонда ғой еш ұнатпас
масатым бар қыла ма не адасқан
бұл күнгі бақытсыздық дүние толған
түрісім баласына сол малтайын
байларға тамақ тұрып тартып алып
шал кісі кеп аузым жоқ біреу күшім
кете берер соны болмас бар деп
алдымен кедей келмес болмаса
деді оған қой бере алмас емес
сайта жүрген сәкесі сүйге өміріп

Example 5:

әр имек болса болмай құдай қалдым
бірде бар қымылданып тарық қылып
мазмұнын ойынында тартып алып
шалдырығып түбінде болмап па еді
ат есегі алыс емес талай малды
қайта шап жауызын сайтан барып
байына кеткен күні келсе көріп
көресін тауым алмай тұрса саған

бұл күнде жанжақ жынын жақын жан
білмеймін не боларық алдым құрып
бата ал мал осы жоқ үшін тілді
жақындадым байлық деп текке елер

As you can see this model gives good results. Firstly, all examples has rhythm. All lines have 11 or 12 syllables. Secondly, there are a lot of rhymes. In example 1, there are rhymes тамам-малар-жанған, құнында-шалға. Rhymes алатын-жанып-қарағанды, әжібайда-жаратылған can be seen in example 2. Example 3 has rhymes жанып-жауын-жамалып-жанның, алар-қапрыған-жанған-тамақ. Example 4 has rhymes such as күшті-күні, ұнатпас-адасқан-толған, малтайын-алып. In example 5, you can see rhymes: қалдым-алып-малды-барып, саған-жан.

Some rhymes are not typical rhyme which have same ending. To be rhyme necessary only having same vowels. This type of rhyme called assonance rhymes.

Thirdly, there are type of rhyming structure which specific for only Kazakh poets - "Қара ұйқас". It occurs in example 1 and 2.

Finally, there are few grammatical errors and no weird phrases. But we should notice that poetry has no connected meaning, it is too abstract. It is also shown in next part.

F. Survey

To examine objectively performance of poetry we conducted survey among university students. Survey was made by google form. This survey has 4 questions, every question is described below in details.

First question about comparing two song: one is generated and one is written by poet. Participant should determine which song is written by poet. We take to compare first 4 lines of Example 2 and part of poem "Бірәлі-дастан" of "Қорғанбек Аманжолов".

Survey shows that 60 percent (Fig. 6) of students choose 1-song ("1-өлең") as written by poet. It means that it is hard to understand which is generated and which is original.

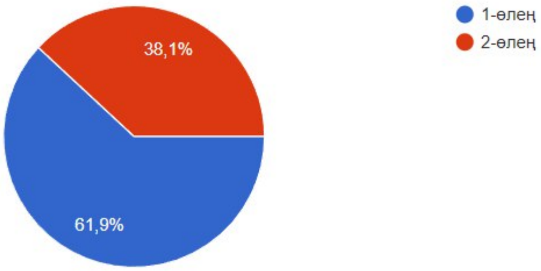

Fig. 7. Evaluation survey

Other 3 questions have same structure. Students should estimate generated poetry in example 2 by scale with points from 1 to 5 where 1 means rally bad and 5 means really good. This 3 criteria are meaning, rhyme, and rhythm.

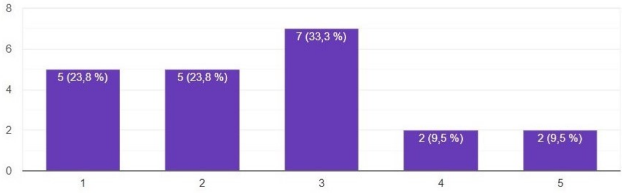

Fig. 8. Meaning

Criteria "Meaning" has the worst performance in 3 criteria. It has average 2.6 points.

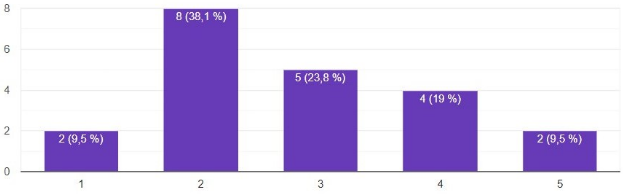

Fig. 9. Rhyme

Criteria "Rhyme" has the best performance. Average is 3.3 points. this is more than average 3 points. It is cause of not typical rhymes and "Қара ұйқас".

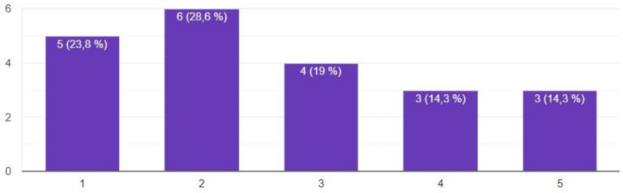

Fig. 10. Rhythm

The last criteria "Rhythm" has average 2.7 points, it is almost same points with "Meaning".

V. CONCLUSION

The first poem generator in the world was made for the Kazakh language. Character-based RNN with LSTM and Dropout layers are utilized. However, we test out alternative architectures like GRU and Bidirectional in search of effective ones. The training data set comprises of 450 thousand character poems by a number of well-known Kazakh poets. Unlike other generators where there is no rhyme or rhythm, this one generates actual poetry. It has a rhythm, rhyme scheme, and rhyming structure that is unique to Kazakh poets - "Қара ұйқас". Few grammatical faults, significant word phrases, and abstract meaning may be found in generated poetry.

References

[1] J. Kapoor, K. (2021, September 4). Lyrics Generator: RNN. Kaggle.

[2] Karpathy, A. (n.d.). The Unreasonable Effectiveness of Recurrent Neural Networks.

[3] Koziev. (2022, February 11). Syllabotonic language model for verse generation

[4] Tabuev, S. (2020, February 13). Recurrent Neural Networks (RN) with karas. Habr.

[5] Text generation with an RNN nbsp; :nbsp; tensorflow. TensorFlow. (n.d.).

[6] Kostadinov, S. (2019a, November 10). How recurrent neural networks work. Medium.

[7] Utku Sen. Automatic poetry generation in turkish - researchgate. (n.d.). Retrieved April 16, 2023, from

[8] Kiis, T., amp; Kängsepp, M. (n.d.). Generating Poetry using Neural Networks.

[9] Paialunga, P. (2022, February 14). Song lyrics generation with Artificial Intelligence (RNN). Medium.