Tanel Kiis
Markus Kängsepp

Generating Poetry using Neural Networks

Abstract

In this work two different approaches to generating poetry with neural networks are described, implemented and tested. Recurrent networks can generate text either word by word or letter by letter. Another approach is using variational autoencoders with recurrent layers in the encoder and decoder. With no available preprocessed poetry datasets available, some basic web scraping techniques were used to collect and clean the training data.

Introduction

Humans have written poems for centuries, and some of us have been able to write great poems and even whole books filled with poems. However, is a computer able to create new poetry by learning from poems written by various artists.

From machine learning standpoint, this task is purely unsupervised. Neural network architectures, that belong to the autoencoders family, are used in this project. Although convolutional layers have been successfully used for sequence learning, recurrent networks are the most intuitive way to deal with natural language data. For this reason, this project focuses on recurrent networks.

In this project, the poems are generated using two completely different methods. The first choice is recurrent variational autoencoder (VAE). The other one is long short-term memory (LSTM) recurrent neural networks, which is used to generate poems word by word and character by character.

Firstly, an quick overview of methods, that have been used previously for sequence generations, is given. In the second section the data collection procedure is described and also the architectures for networks used in the project. In the final section we summarise the results and discuss possible ways to improve upon the work done so far.

Background

There are few articles and blog-posts about poem generation using recurrent neural networks. Xingxing Zhang and Mirella Lapata generated chinese poetry using recurrent neural networks [1]. They predicted next character based on previous character and the context of previous lines. As a result, it outputted high quality poems, comparable to the state of the art, although not reaching human level.

Peter Potash et al. did not exactly try to generate poems, rather rap lyrics, more precisely novel lyrics in style of target artist [2]. In the article special token for line- and verse end was used, so network could learn the structure. The model was able to outperform previous implementations.

One blog writer used LSTM recurrent neural network to generate poems [3]. He used haikus and limericks for learning and the network was able to produce these kind of poems with correct structure, however not consistently.

In these articles, character by character method was used, nevertheless articles about generating poetry using subwords was not found. In this project, both the character-based and subword based models are tried out.

Variational autoencoders have also been used as method for generating arbitrary sequences. In this project a method described by the Google Brain Team [4] is replicated. Although they used it for generating sentences, the same ideas can be used for any sequence.

Methods

Data Collection

Most of the poems were collected from two sites: Poem Hunter [5] and Poetry Foundation [6]. We were able to extract around 150 000 poems written by top 500 poets. In the cleaning phase poems that were not in english were removed. Some authors have written several thousands of poems - to avoid being biased towards them, the number of poems per author was limited to 500. As a result of these steps, only 50 000 poems were kept for the training.

Furthermore about 4300 haikus were collected from tempslibres [7] Haiku database. It is a bilingual site and most haikus have english and french versions. Both of them are presented in a single column with no language reference attached to them. A simple language detector was built using common word lists for both languages and only haikus written in english were kept.

Data Cleaning

For usual character by character approach only english alphabet letters and main punctuation was left in, total of 80 characters. Before cleaning there were 1234 different characters in use, mostly because some poems with hieroglyphs were undetected previously. Obviously other poems with many unused symbols were left out, there were 194 such poems. In the end there was 83 million characters worth of poems, which contained 15 million words.

For subword approach the data was first converted into lowercase and tokenized, last means adding whitespace around punctuation characters. After that byte-pair encoding was used to split the data into subwords. In this project vocabulary size was reduced to 30 000 subwords, compared to 180 000 different words.

Haikus have a fixed structure - 3 lines per poem. This makes it rather easy to clean them. After removing html tags, punctuations and excessive white space, we ended up with almost 7 000 unique words between all of them, out of which less than 4 000 had more than one occurrence.

LSTM

LSTM stands for long short-term memory and it is building unit for layers of a recurrent neural network. The LSTM unit is made of a cell, an input, an output and a forget gate. These are responsible for memorizing over certain time period, where the gates regulate how much of the data is kept.

There were two main approaches for training LSTM models for poems. Firstly character by character, it means that models get character sequence as an input. Based on the the first character the model tries to predict the following one, next based on first two it tries to predict the third character and so on. Second approach was subword by subword, which was essentially the same as character by character - the model tries to predict next subword based on previous.

Furthermore there were used two ways to split the data, the obvious one is poem by poem and other one would be based on specific sequence length. For poem by poem approach the poems have to be made into equal length, so the model could use batch updates. However, this is going to increase the training time, because there is more data to go through and the lengths of poems vary a lot. On the other hand, the limited sequence length will probably lose some context of the poem, because the network does not see the poem as a whole. Nonetheless both of these approaches were implemented and tried.

Last but not least, generating poems with the LSTM model. The model starts from one random or inputted character and predicts from that until end symbol or length limit is reached. Based on the temperature, the most likely character or character from predicted distribution is selected. So if the temperature is close to 1 the output is more likely to be randomly chosen based on the predicted distribution. On the other hand, if the temperature is 0, the most likely character for next spot is chosen.

Variational Autoencoders (VAE)

Just like regular autoencoders, variational autoencoders have a symmetric structure, with smaller layers in the middle, to force the model to generalize instead of memorizing the mappings. The only difference between these two is the middle layer. In VAE the middle layer does random sampling from several gaussian distributions. For this the encoder network outputs two vectors of equal length - one for the means (μ) of the distributions and other for the variance (σ^2), as seen on figure 1. Now a random vector $x \sim N(0, 1)$ is sampled and the latent vector is computed: $l = \mu + x\sigma^2$. This latent vector is now passed through the decoder part of the network.

The loss function is a combination of the difference between input and output layers and the Kullback-Leibler (KL) divergence of the middle layer and a normal distribution. The KL divergence is a common similarity measure for distributions. It is required for two reasons - firstly it punishes the VAE, when it tries to turn itself into regular autoencoder by making the σ always equal 0. And secondly, when the distribution of the middle layer is approximately known, it can be used for generating new outputs, by sampling from the same distribution and applying the decoder to the newly sampled vector.

For sequential inputs, instead of regular fully connected layers, we have to use recurrent layers. In this implementation the gated recurrent units (GRU) were used, because it is said that it has similar accuracy at better performance. Most of the layers between the input and the middle layer output a sequence, just the final one will return only the last vector, that will be converted to the mean and variance with fully connected layers.

The decoder can use recurrect layers similarly - the latent variable is the initial state for the first recurrent layer and all of then return sequence. The last layer will be softmax over all possible words.

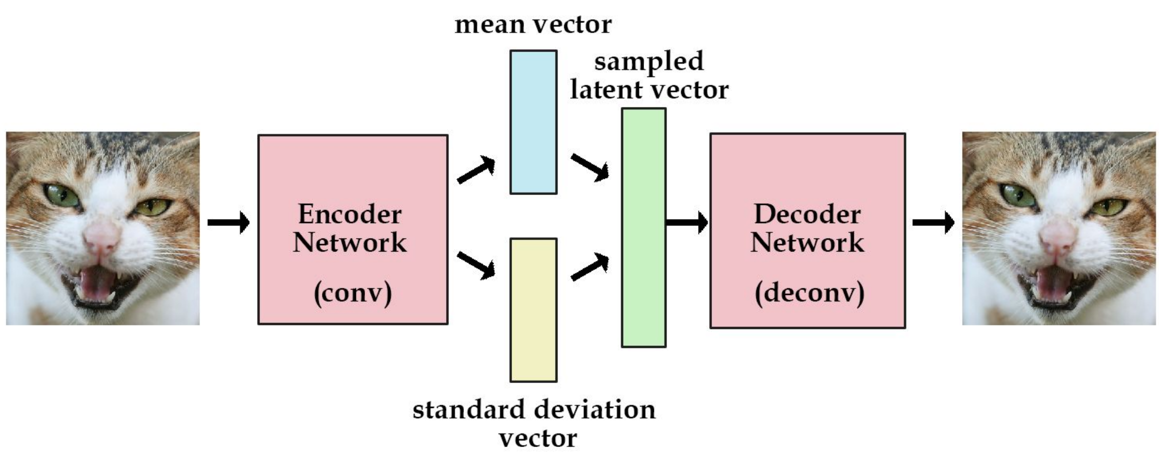

Figure 1: Basic structure of a variational autoencoder [8]

Results

LSTM - Haikus

Models for haikus were trained with poem by poem approach only. First the results of generating haikus character by character. For this a model consisting of 2 layers and 500 hidden units on layer with 0.2 dropout was used. In case of generating haikus the temperature around 0.6 gave best results, because there were less random characters and broken words. Overall, the results were not very good, the model was not able to learn the structure of haikus very well. It used short sentences, however it did not output end symbol at third sentence, so just first three rows were taken. Next there are some good unedited haikus, although most of them did not make much sense.

Haiku 1

rose back
from a three whisks
at the embers

Haiku 2

daybreak
a cold snow falls
into the sound of the bay

Next generating haikus with a model trained on subwords, there were about 7000 different words, although only 4300 haikus. The structure of model was same as for character by character approach. In case of subwords the outputted haikus made even less sense. This happened probably cause of small amount of training data. Next some unedited examples.

Haiku 1

unsigned of fly
of the butterfly blue
the moon of a valley

Haiku 2

nightfall chill
white breath into a hangs
of the sound of my haze

Tanel Kiis
Markus Kängsepp

LSTM - Poems (Character by Character)

Models for usual poems were trained using different sequence lengths from 100 to 400. Also different dropouts and number of layers were trained. Poem by poem approach was also tried, but it did not give good results for some reason. Nevertheless, all of sequence-based models gave rather good outputs, with fluent english, correct punctuation and poetic structure. In addition most of the words were actual english words, but due to the random nature of the generating, some of the them were not. Furthermore most of the poems lacked overall meaning and uniform context. Next there are handpicked unedited outputs from different models for character by character approach.

Poem (500 hidden nodes, 2 layers, 300 sequence length, 0.1 dropout)

Fatherful and broken throne.
The sight of the conqueror seems;
The spring is stiff and strong,
And the small feet of things are still enlightened,
And these his ears of soul in the sun 'sing.
The wind came on to see, he looks back,
In the hills
Where the boys can’t have a stream of space

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
Shadowed to his lost story.

Poem (500, 3 layers, 100 seq, 0.1 drop)

o me warm me
to all its movements of an unknown,
and me the world forced to blood the time,
her fringe in the blazing sound of the wind

I've done conformation for freedom,
the plans shake up my man.
Reminiscent this Visa,
did you forget for
when your whole world will take me
I did so, like a boy
from the plane
the moment was darker
facing him, I felt a reminding cup

from a boot with the absolute, light;
goes gently, so diligrant, and deaf is still going before
there was something about earth

Poem (500, 3 layers, 100 seq, 0.2 drop)

That sad, fancy's absence, it's always through,
Please do abandon from the spanish star,
That is your ill-eyed family by me to fear
You feel, I'll seek out range, no cancer's legend roars to fear,
The crazy world of faith and humans ever steer?
With hand or human thoughts are stricken
Dead sky, an infant in their praise.
And, majestic and loyals and maybe
Besides, yet turn for there;

Poem (500, 3 layers, 100 seq, 0.2 drop)

Caesar bound by the Divine,
I'm not going to be invisible,
What love, I deserve, maybe I"m new,
Nothing more,
And then there isn't they not to teach.
The next city of their attentions lost
Your presences change, to walk the classics of men -
It was my native day that embress of the wind
Rise against thee! Miilling on my shoulder,
And your hair in a red and glowing

LSTM - Poems (Subwords)

In case of subword based generation, the poems had also fluent english, mostly correct punctuation, except for apostrophe usage, and poetic structure. On the other hand the writing style was more jumpy and made often less sense. In addition splitted words are not put together correctly (i.e “heart-shoes”). This occurs probably, because the words that are split into subwords are less frequent, so for the model it is harder to learn correct order of these. It is worth noting that sequence length here means the number of subwords used in one batch. Next some unedited poems generated using subwords with various parameters.

Poem (500, 2 layers, 30 sequence length, 0.2 dropout)

rendered more but move;
young cruelty,
crossed the other lands,

and the lotus wins in the air.
the wind waits for the answers,
the arrange just the old,
and they wait, especially then have a mate.
about you, sleeping and remembering.
the world is all amongst her,
her soul crowded up the world;
yet this florida image!
who will the world walk round?
she is the others to save her
he is the essence of a child.
dear despair,
but she is a jewel

Poem (500, 2 layers, 30 seq, 0.2 drop)

chanting the song,
hold the hand,
and the precious ones that eat,
even more than those in the cases are passions.
oh, how much i feel when i am at home!
'pressing, love and feed
i've thought of you that
when one that brings the impossible to me
is a hurry, even though i'm the truth dear,
i'll understand you,
you will forever be gone.
i live, without you, not you.
heartshoes,
his eyes as old as

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
(perhaps we're all it's meant for us)
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

Overall the quality of poems for both character and subword based models were varying a little, the fluency and grammatics were alright, however the most important part, the meaning was lacking. It is hard to say which parameters or approaches gave better results as all results seemed pretty similar. In the appendix, there are also some randomly picked poems to show the typical quality.

VAE results

With variational autoencoders we had similar results - it seemed to be able to learn some pairs and triplets of words that occur together, but very few coherent poems or haikus were formed.
The haiku dataset was rather small - the network did not have to be very large and there was more time to work on different permutations of hyperparameters. The most interesting was tuning the loss function. Larger importance on the Kullback-Leibler divergence made the convergence slower and training error larger, but in the sampling phase the haikus seemed to make more sense.

sound of this the wind still leaves	all the this the night garden roses for a	the window out of I young of her

Discussion

To improve current LSTM results, networks with larger hidden layers or more complex structure could be tried. In addition, the results really depend on the quality of poems. Hence poems from one author or similar authors could be used for training to see whether it improves the quality of the poems. In addition getting poem by poem approach working properly could give better results.

In order to get final results we had to try different computing cluster, because the main one (HPC) had really long queues at certain periods of times. So next is given some notes of different clusters:

- HPC - it was really easy to get scripts running there, unfortunately often the queue waiting times were really long.
- Tensorport - it was also rather easy to get running. Although for some reason Keras worked there only with Tensorflow version 1.0.0 and Python 3.5, which took some time to figure out. Unfortunately it had only certain amount of free GPU hours.
- EENet cluster - it took couple of hours to get jobs running there, even with introductions. In the end best option for us, as there were no queues for GPU jobs.
- AWS - approved our account in 24h, but it was too big for our needs and we could not manage to find a simple way to run our models on their servers.

Conclusion

In conclusion, two different approaches were used to generate poems. LSTM networks learned the structures of poems and grammatics of english quite well, however lacked the overall meaning. Nevertheless there were some poems that were somewhat reletable. On the other hand VAE seemed mainly to learn some pairs and triplets of words that were common in the training data. In the sampling phase it managed to form some poems that did not exist in the training data and could interpreted as a original and coherent poetry.

Authors Contributions

Implementing LSTM for poems, using code base for character by character text generation. However it had to be changed to fit our needs, for example poem generation with temperature. Furthermore poem by poem training was added, not just sequence by sequence training. LSTM was previously not done on large number of poem types, only for haikus and limericks and for Chinese poetry. In addition there was no literature about subword based poem generation.

Also an recurrent variational autoencoder, that was proposed by a research paper, was implemented, which was used only on haikus, unfortunately it did not work out as well as expected.

References

[1] "Chinese Poetry Generation with Recurrent Neural Networks." http://www.aclweb.org/anthology/D14-1074. Accessed 3 Dec. 2017.
[2] "GhostWriter: Using an LSTM for Automatic Rap Lyric Generation." http://www.emnlp2015.org/proceedings/EMNLP/pdf/EMNLP221.pdf. Accessed 3 Dec. 2017.
[3] "Generating Poetry with PoetRNN - Machine Learning by Doing!." http://sballas8.github.io/2015/08/11/Poet-RNN.html. Accessed 3 Dec. 2017.
[4] "[1511.06349] Generating Sentences from a Continuous Space - arXiv." 19 Nov. 2015, https://arxiv.org/abs/1511.06349. Accessed 3 Dec. 2017.
[5] "Poem Hunter." https://www.poemhunter.com/. Accessed 11 Jan. 2018.
[6] "Poetry Foundation." https://www.poetryfoundation.org/. Accessed 11 Jan. 2018.
[7] "Haiku - tempslibres : database of haikus." http://www.tempslibres.org/tl/en/dbhk00.html. Accessed 11 Jan. 2018.
[8] Kevin Frans: “Variational autoencoder explained”. http://kvfrans.com/variational-autoencoders-explained/. Accessed 11 Jan. 2018.

Appendix

Randomly picked poems with various parameters.

Haikus (characters)

Yeasan pannes
do mushiat moon glass
the holes against

morning says
grows aottored
a parsy feets

night fires
down traffic
forst bottles river

Haikus (subwords)

Shivering frost
a caterpillar blue
leaf in the jardin

godless a through
ice in the valley
log morning winter

trunk fog
the holes of blue stones
sit to a fog

Poem (500, 2 layers, 300 sequence length, 0.1 dropout, characters)

outeoughs well,
He lies like the grasses of the clear,
Amazing hib, his face and game,
As when the sun were stiring around?
A time for a bit on some build.
Your float make wards of basil pase,
To surviver, listen, to disanew.
Jay Jesus dream dies laugh; but dyne
Man wenty snares gray to the earth,
Life in the sea Dogsmairue world with Emorious Good.

Just the sustend standers of half Daughters!
Goodbye's Absorraritory of Maintainant,
Against the Sylvan Heavens, the pork of slaughter,

Poem (500, 2 layers, 200 seq, 0.1 drop, characters)

in his golden height
That will fear me, I am another other...oblia, wisdom self shock to you.
I found it all their manners you"ll see
your mind is-
And you saved, when I was my manian.
What all have been, then you fall so own a joke?
People want to labor you in their trail and love not to give you feelings:
and worse "Her 'Petty? I real youth
..) Yea, if it has stand, ere beauty took your arms, eyes don't believe that sudden, you"ll
jester struggle - never did you retire

Poem (500, 2 layers, 100 seq, 0.2 drop, characters)

Might and be For Other.

We must find their lips
In an insects
Drug unchanged in love.
Knowing what sun fenct on him totally.
(Don't believe anything"
Would it be)
Is the true excess
And wompary in rays
Depression repays?
i won"t call me
A life of join
For that you love and say
Your answer keeps it

Poem (1000, 2 layers, 30 seq, 0.2 dropout, subwords)

brighted time and life delight.
we saw those thousand frogs but now,
blue as spring's shipwreck,
tall and mounts the couple wait,
birds no longer come,
nor longer shall guest speak,
to nurse in empire bridal tti .
the younger dogs sleep,

constellation heavy as she is:
"avarice - tout ses with the might of silence,
lights and arts into her darkest grave,
tests each, "
tis a good day for meet.
death and alarm must be the same,
a worthy gun, and death

Poem (500, 2 layers, 30 seq, 0.2 dropout, subwords)

vet and the impotent greed
and spiders grown wrong
and most special clue
the physical female frame fail
perhaps for circus
nice girls with immense harmony
find then again,
passing much for some of us
all ist and each other dig
and nectar herself has no apparition of my ahead
or to be cakes we're without the ones
so much cute flower - nectar
smoping your hero tomorrow,
breathe with you accumusystem
begin to fan younger - kisses in a primibluesoul's
also in this second in the day morning.

Poem (500, 2 layers, 30 seq, 0.0 dropout, subwords)

washed to death without straight but
that no ill - rich man tossing under the sphere
he does know of us lives and hurt
god will for him follows us away
we fall hollywood up
all we will in this way
also let him speak of bended drugs and comfort him,
and our honor we will always be
the will of only rights 17nib
the mom sathasivam asks him about you
when day beautifully blows thru
all he wanted this to sleep with him
and the wolf in the night were alive
you're only in one