Project Report

CS 725 - Foundations of Machine Learning

Team Members

Anmol Mishra	150010041
Utkarsh Kumar	130050022
Anand Dhoot	130070009
Maulik Shah	13D100004

Generative models are used for modelling latent features in a dataset and performing inference tasks using the Bayes' rule. In addition to this, these models can be used for interesting predictive tasks such as generating text from the language model learnt.

For this project, we consider the problem of learning generative models for poems. We aim to explore generative language models using Recurrent Neural Networks, these are neural networks with loops, allowing information to persist between several inputs. We investigate the application of such models to datasets of poems.

Related Work

Generative models are not uncommon. One can find several implementations for specific language models based on RNNs - cooking recipes, composing music, Linux source code, writing compilable LaTeX code and even rewriting the Bible! The results on these datasets show that the model is able to learn several high-level features (the model for Linux source code could capture indentation, appropriate opening and closing of quotes and brackets, etc accurately).

By the time we started this project, no implementation of language models for poetry existed, to the best of our knowledge. Is this because such models do not have sufficient power to capture the essential characteristics of poems? We find out in this project!

In addition, we also seek answer to the question: can we create language models to capture the inherent structure present in most poems? A little ambitious question would be, can the rhyme scheme of the poem be captured?

Problem Statement

We were curious to explore the techniques involved in training RNNs and specific implementations such as LSTMs taught in the theory lectures. We wanted to learn about the power of Recurrent Neural Networks and compare the prediction outputs to find out how much structure they are able to capture from the training text.

Since there have been no efforts in this direction, we had to handle other auxiliary tasks as well, such as identifying appropriate datasets, collecting & cleaning data, creating & tuning the model and analysing results (there are no known metrics for quantifying the quality of poems), each of which are described in detail below.

Data

Types of Poems (source)

- Sonnet - a short rhyming poem with 14 lines. Found and developed by writers such as Shakespeare. Sonnets use iambic meter in each line and use line-ending rhymes.
- Limerick - a five-line witty poem with a distinctive rhythm. The first, second and fifth lines, the longer lines, rhyme. The third and fourth shorter lines rhyme. (A-A-B-B-A).
- Haiku - ancient form of poem writing. Haikus are composed of 3 lines, each a phrase. The first line typically has 5 syllables, second line has 7 and the 3rd and last line repeats another 5.
- Epic - a lengthy narrative poem in grand language celebrating the adventures and accomplishments of a legendary or conventional hero
- Couplet - two lines of verse which rhyme and form a unit alone or as part of a poem
- Narrative - A narrative poem tells the story of an event in the form of a poem.
- Free Verse - A Free Verse Poem does not follow any rules. Their creation is completely in the hands of the author.

A glance through few poems of each type indicated that Sonnets have a well-defined structure to every poem - rhyming, syllable count, length of the poem & individual lines. Haikus also have a structure associated with them. However, these were written in other languages and the linguistic features were lost in the process of translation to English. While Limericks have a rhythm, they lack other structural features. Hence we decided to stick to sonnets for our project.

Preparing the dataset

We looked through several sources in search of sonnet dumps. We found a few websites that hosted a small number of sonnets for a few poets. Since the writing style could vary significantly with authors, we decided to locate a collection of a large number of sonnets written in a similar style. In addition, we wanted an online collection having well structured URLs for fast scraping.

We found a good corpus of poems at this website. It has poems written by several renowned poets and for the purpose of our project, ~1400 sonnets written in Shakespearean style . Most other websites either didn't have enough sonnets or did not have data in the structured format.

We wrote Python scripts using *BeautifulSoup* to scrape his poems from this website and the dump was cleaned to obtain just the text of the poem, removing HTML elements, headers and footers, etc.

Our best performing solution

We train a character-LSTM on the Shakespeare-sonnet corpus obtained as described above. The model had 2 LSTM layers and was trained for 1000 epochs. We saved the model every 100 epochs to be able to analyze the kind of text that it could generate.

The final model captured some rhyming schemes, evident by the ending syllables of each line. The overall structure of a sonnet consists of 14 lines, most of the outputs were close with 13-18 lines for each poem.

The predictions of our model and their analysis are discussed in the results section below.

Alternative approaches explored

Training on dump of all (types of) poems

Our first approach to this problem was to scrape all poems from a website and simply train our Recurrent Neural Network on the dump. Unlike Shakespearean sonnets, these poems did not have any structure or rhyme scheme to it.

As expected, we could not learn a ‘good’ language model from this corpus. The lines in the generated poems were of varying lengths (sometimes as long as 128 characters!). Additionally, the training corpus had poems from several authors and therefore the Neural Network could not learn any one style perfectly. Some outputs generated are in the section below.

Vanilla RNNs

In general, LSTMs are known to be superior in terms of generated content as compared to RNNs. This was pretty clear through the difference in the structure of the poems generated in the two cases.

This can be explained through the issues of vanishing gradient and exploding gradient that RNNs suffer from. LSTMs overcome this issue using gated logic for writing to and forgetting from the cell state, thus enabling them to “remember” information longer.

Syllable-level Recurrent Networks

Very often, we think of poems as a bunch of lines that rhyme! Moreover, since we use sonnets as training corpus, we were curious to find out if syllable RNNs would result in learning of ‘better’ language models.

But like the case with most other parts of this project, there is no existing implementations for breaking words into syllables. There are some tools such as CMU Pronunciation Dictionary *cmudict* which converts known words into a phonetic respelling of the word. Even if we could train a syllable RNN, converting the generated text back to normal English would be another challenge.

Also Engligh language is known to have a whooping ~10,000 syllables. Therefore we would have to implement a way to reduce the dimensionality of the space of syllables for better performance.

Results

Our network generated the following sonnet(s) after being trained -

Thy depplely, but is, innouses trreent!--"Old
Its ips the Mistay acly tardered cure
Of lips from that yef time may the lood--
Verece instepied night, smeanow bitter ser
Wap think-plowers then dope here, and kest-tanst "Ir as can shall trine,
And at morn throng recolses her ever and wife,
The will not, and Whose pulse helf at 'ty even,
In, that whoth it shall it winding sprend;
Our living she win tearms away, while thou bless,.
Decunnish sweet in her ard it set thee;
The ralish's dush out with the
Spirit him enterne propped to the pain life;
And o'er Soon, may the onith. Lorl of wonder then
With then some to. Lord, seems of viling swizberms.

While the sentence semantics are not good, at least consider the choice of vocabulary. In fact, if you do an emphatic reading, the generated poetry does have a pseudo-intellectual feel to it - something that one can't make sense of in the first reading, but which gives an expression of deeper meaning, kind of how art is supposed to be!

Results of some of our other experiments are in the sections below.

Training on all types of poems

As described earlier, we trained our first model on a dump of all types of poems by a variety of authors. Here is a snippet of the generated output:

Untile you my work in a-bleeding slow
And wished him hear:
Heard listenmen subtil
A word will be not leave you, and the blended fashions Patria superb Mary came,
And other wave and nature's site of man, that stares,
In peace, the faculties, God he in a far.
"Yank age is leaves,

fa-scrabity,
Long roses—. . . .
O the timid poems like it to Sunset along these Or a victim.
it’s one with lilies-channels from love
Here as far of Heaven and everything to them that timed every gladness and canoe! 'that' Enough, though thy phones have a summer's long;

Variance of outputs with training epochs

In this section, we showcase how our model started learning the sonnets. We present few lines of the output generated by the trained language model when trained for 100, 200, 500, 1000 epochs.

100 epochs

Ancient sects of yet, onror slight.
But upunlinic, with some the sayward in smarch;
The storwing her shillent dim yet dray,
There's golden siance, in the wastery-day

200 epochs

And tromas that brown frost hails of aught,
We woods learms from the gathers day-things, old breal
And wently wither yet these surewing love
Worn I seak light this Brights the do oncine--

500 epochs

And gone of plush of mystoor) cast that,
With up there whose peffore a must divine deep
Of the speen to decunlous deep least,

1000 epochs

Our living she win tearms away, while thou bless,.
Decunnish sweet in her ard it set thee;
The ralish's dush out with the

Clearly, the model is improving across epochs and is learning better and complete English words with correct spellings get predicted.

Vanilla RNNs vs LSTMs

When trained on LSTMs, we observe (above) that the output generated is very structured - the lengths of most lines are about the same, the size of each poem is 14-15 lines, etc. When trained on vanilla RNNs, we got the following output, clearly indicating the improved performance of LSTMs over RNNs.

Some courle
Thy depart you cannot pharmory
Springs;
Dost kissled at block dream,
Where a death gave all her street
On this pure of Green collowing the starry spaceful hand that still is music thirmulled his midswed his wilder sheen!
The for chank onsels surpwed Round's dust; how away of was once solitude.
Yet way he presence?
The self with more through haudeland'd was short, where it a died, thee.

That the dead,
A cily,
Such the passion distancing might his name look, tired in the world as a worn,
Old you beyond

References

[1] Classification of poems into types
[2] Collection of Shakespearan sonnets
[3] Andrej Karpathy's blog on Recurrent Neural Networks
[4] Torch implementation of RNN by Justin Johnson