1 Introduction [0 points]

- Group members:
  1. Sri Aditya Deevi
  2. Palak Purohit
  3. Princekumar Kothadiya
- Team Name : Rhyme Robot
- Colab link:
  - HMM Based Model
    https://colab.research.google.com/drive/1NYl_RasUF-bYUVZKDcnJiKVQ5nSxrH_R?usp=sharing
  - LSTM Based Model
    https://colab.research.google.com/drive/1PmC9vrkuaudILyggQ3akTxbkeIy8Ndx8
- Piazza link:
  https://piazza.com/class/lbv0docn6037fw/post/734
- Division of labor:
  - Basic Plan of Work* : Sri Aditya Deevi, Palak Purohit and Princekumar Kothadiya
  - Data Preprocessing and Syllable Restriction : Sri Aditya Deevi
  - Generating Entire Poem in One Go : Palak Purohit
  - Line by Line Generation : Princekumar Kothadiya
  - Poem Postprocessing : Sri Aditya Deevi
  - Incorporating Rhyme* : Palak Purohit and Princekumar Kothadiya
  - LSTM Based Model : Sri Aditya Deevi
  - Visualization Code for Analysis : Palak Purohit
  - Analysis and Inferences* : Sri Aditya Deevi, Palak Purohit and Princekumar Kothadiya
  - Piazza Post * : Sri Aditya Deevi, Palak Purohit and Princekumar Kothadiya
  - Report and Colab * : Sri Aditya Deevi, Palak Purohit and Princekumar Kothadiya
- Packages used:
  - Numpy
  - Matplotlib
  - hmmlearn
  - WordCloud
  - Regex

*Equal Contributions

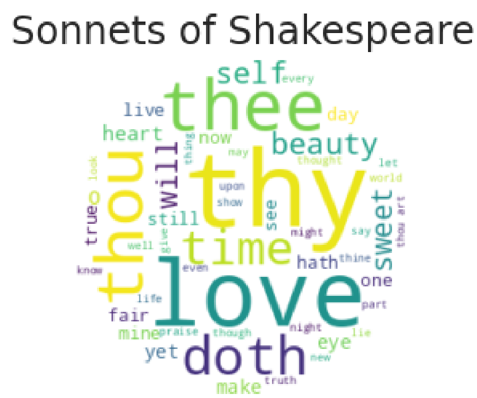


Figure 1: Wordcloud Representation of Shakespeare's Sonnets

2 Pre-processing [15 points]

We followed different schemes for preprocessing the data for the HMM and LSTM based models. In this section, we describe these schemes.

For HMM Based Models

We analyzed the text data given and found it has both upper case and lower case alphabets along with the following unique non-alphanumeric special characters:

```
{'!', "'", '(', ')', ',', '-', '.', ':', ';', '?'}
```

So, we first removed the following the symbols Exclamation, Brackets, Comma, Full Stop, Colon, Semicolon and Question Mark as they might make it complicated for the HMM model to grasp the main content of the poem. Also, these characters can disrupt the flow of the generated poem by appearing in wrong places and/or consecutively. But we had to careful considering about how to process hyphens and apostrophes (single quote symbol) as just dropping them might make some words meaningless. Also, the capitalization was removed so that the model can focus on actual semantics (word transitions etc) rather than the other aspects.

After such preprocessing steps, we made sure that the number of unique words in the syllable dictionary match with the number of unique words in the sonnets. This would enable us to produce syllable accurate poems, as we will discuss in the later sections. After we finished the tokenization, the tokens were numerically encoded with integers. Then for getting the training sequences, we considered two different schemes (Discussed Further in the next sections) : Line-by-Line Generation and Generating the entire poem in one go. For the first case, each training sequence is the encoded version of a line whereas in the other case, each training sequence is the encoded version of a poem.

For LSTM Based Models

As suggested, we planned to implement a character-based LSTM model. We considered two different schemes for preprocessing. In the first case, we considered text data with captitalization and special characters. The number of unique symbols in this case was 61. In the second case, we considered text data without captitalization and but with special characters. The number of unique symbols in this case was 38. The intention behind this to study the performance of the model in terms of semantics in both the cases. Then we numerically encoded the data. Finally, for training we consider sequences of fixedlength(40 characters). We do this by taking subsequences of 40 consecutive characters from the dataset. Also, batching and one hot encoding of input data is performed.

3 Unsupervised Learning [20 points]

In this project, we are considering a first-order hidden Markov model (HMM) to model the joint distribution $P(x, y)$ and we are in the domain of unsupervised learning, where we are given a training set of N training examples containing only the $\mathbf{x}$'s :

$$S = \{\mathbf{x}_i\}_{i=1}^N$$

Note that each $\mathbf{x}_i$ is a sequence. So basically get the maximum likelihood problem to be:

$$\operatorname{argmax} \prod_{i=1}^N P\left(\mathbf{x}_i\right) = \operatorname{argmax} \prod_{i=1}^N \sum_{\mathbf{y}} P\left(\mathbf{x}_i, \mathbf{y}\right)$$

Then we can converge to the optimal parameters (Transition and Emission Matrices) using the Baum-Welch Algorithm whose main steps are as follows:

(i) INITIALIZATION:

Here we randomly initialize HMM model parameters.

(ii) E-STEP:

We run Forward-Backward algorithm to compute marginal probabilities $P\left(y_i^j = a, \mathbf{x}_i\right)$ and $P\left(y_i^j = b, y_i^{j-1} = a, \mathbf{x}_i\right)$ based on the current model parameters.

(iii) M-STEP:

We then do a soft update of the model parameters based on the maximium likelihood estimate given the data as follows:

$$P\left(y^j = b \mid y^{j-1} = a\right) = \frac{\sum_{i=1}^N \sum_{k=1}^{M_i} P\left(y_i^k = b, y_i^{k-1} = a, \mathbf{x}_i\right)}{\sum_{i=1}^N \sum_{k=1}^{M_i} P\left(y_i^{k-1} = a, \mathbf{x}_i\right)}.$$

$$P\left(x^j = w \mid y^j = a\right) = \frac{\sum_{i=1}^N \sum_{k=1}^{M_i} \mathbf{1}_{\left[x_i^k = w\right]} P\left(y_i^k = a, \mathbf{x}_i\right)}{\sum_{i=1}^N \sum_{k=1}^{M_i} P\left(y_i^k = a, \mathbf{x}_i\right)}.$$

(iv) If we do not converge yet, we can repeat from Step 2.

We tried two different ways of methods of generating the poem:

1. **Line By Line Generation**:

   Here, the training sequences are different lines in all the sonnets.

2. **Generating Entire Poem in One Go**:

   Here, we treat the entire poem as a single training sequence.

For both cases, the model was trained using `hmmlearn` package since we wanted to experiment with a higher number of hidden states and the package is highly optimized which enabled quicker training. Further details about the training process for both these models are as follows:

- **Line By Line Generation**:

  Model from hmmlearn has been trained using 100 states and 1000 iterations. After training, transition and emission matrices has been saved to generate poem using helper code from homework with few modification of function to generate fixed syllable.

- **Generating Entire Poem in One Go**:

  The dataset consisted of a list of lists wherein each inner list consisted of the sequence of tokens from the whole poem. The number of hidden states was chosen to be 8 as that would be sufficient to encompass all parts of peech such as nouns, adjectives, verbs, etc. Moreover, since the dataset is small, training for larger hidden states could lead to overfitting.

4 Poetry Generation [20 points]

Schemes of Generation for the Poem

Since the we are considering entirely different approaches so we considered slightly differnt ways to get the final poem. Following are the details of the output from both these HMM models:

- **Line By Line Generation**:

  Total of 14 line has been generated using the trained model one-by-one. The total of 10 syllable were generated for each line. The syllable dictionary has been used to keep track on syllable count which also considered the cases of different syllable count for single word while generating. Here, each line of the poem has been generated independently. Helper code from the homework was modified and used for the syllable generation. The line-by-line generated poem is grammatically more accurate although the theme may not be consistent since all generated lines are independent of each other.Still, It has been generated as Shakespeare's style with fixed 10 syllable in each line so, it has some meaning. One example using this method is as follows :

```
mine best is at thou art whoe'er not should
another's our by her seen delight
part ever not help sun's from lie of truth
treasure no love we her a treasure not
spite thee that with day to i that to fair
my other shall come contend to more the
sin despised or truth world die you such
were through all edge praises her for doth seem
praised be ranged sick their on and thee sight might
love with love thy recompense worse born not
to bare thrall but spite leave bear i staineth
may dear if where that knowledge mansion though
perfection even ne'er thou each alone it
not growth show monument hide day truth face
```

- **Generating Entire Poem in One Go**:

  For the generating process, the first word was chosen randomly while the remaining words in the same line were chosen such that the total number of syllables in the line amounted to 10. This was done using the syllable dictionary, just like the line-by-line generation. The difference was that the first word of the next line was chosen according to the last word of the previous line by using the A and O matrices. Helper code from the homework was modified and used for the sequence generation.
  The poems generated in this manner were theme-based, had a consistent flow and they retained Shakespeare's style. Although they weren't grammatically sound, the theme was maintained within the entire poem. This could be because since word sequences from an entire poem were given in the training set, HMM learnt the flow of thought but the dataset wasn't aptly large for it to learn the grammatical nuances as well. One example of the poem generated by this model is given below. As discussed, the entire poem revolves around the concept of the death of a loved one and being self-reflective and satisfied with one's own journey as the time of one's end nears.

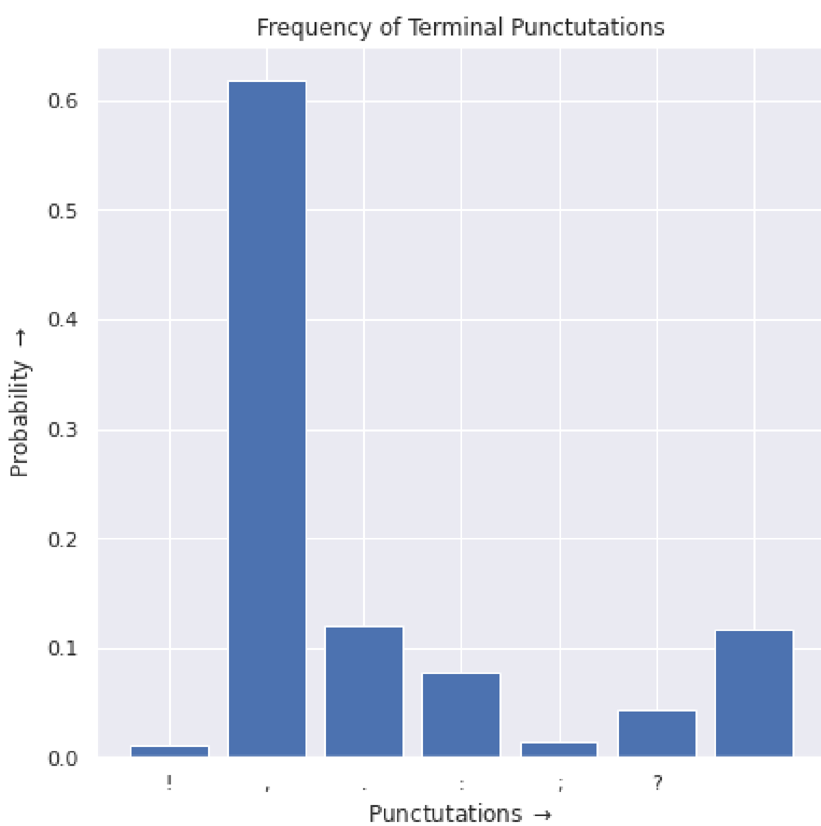
```
for disease dying virtue my grief disabled
betwixt do such make home even so his
in joy hath the because worthy beloved
but on youth substance which upon hath my
sport but tied thou when time and poor heart but
```

```
be made with that asleep thou be same worths
commits faith do tomb thine blame bases death's
and where part thine though cloud but it doth the
thou like vantage is journey that sweet fed
when lay fleet'st of time war to that your he
the it better may though me in is thus
tell evil am thee be end true bound but
till comments you had brand spirit in himself
up dost only sequent heart gain acquainted
```

Capitalization and punctuation were later added to the generated sequence using post-processing, described in the next subsection.

Postprocessing the Poems

Postprocessing of the poem mainly included Capitalization of the first letters of each line and addition of punctuations or special symbols at the end of each line. So first we considered the frequency of occurence of all the terminal punctuations as follows:



Then based on this probability distribution we randomly sampled one of the terminal punctuation at the end of each line. One of the example poem after post-processing is as follows:

```
Mine best is at thou art whoe'er not should.
Another's our by her seen delight
Part ever not help sun's from lie of truth:
Treasure no love we her a treasure not.
Spite thee that with day to i that to fair
My other shall come contend to more the:
Sin despised or truth world die you such.
Were through all edge praises her for doth seem;
Praised be ranged sick their on and thee sight might!
Love with love thy recompense worse born not,
To bare thrall but spite leave bear i staineth
May dear if where that knowledge mansion though,
  Perfection even ne'er thou each alone it,
  Not growth show monument hide day truth face
```

5 Additional Goals [10 points]

Incorporating Rhyme

To included rhyme scheme *abab cdcd efef gg* in the poem, we selected the line-by-line model since it would be easier to generate rhyming. We made the following changes to the line-by-line generation algorithm:

- The model was trained on reversed word sequences so that it learned to generate lines in reverse order.
- A list containing sets was created such that each set contained words rhyming with each other. This was collected by forming tuples from the rhyming words in each sonnet, according to the rhyme scheme (except sonnets 99 and 125 since they are not actual sonnets) and then pooling all tuples with common elements into sets.
- 7 sets were chosen at random and 2 words were chosen from each of these sets
- Lines were generated in a reverse manner by setting these words as the seed for starting the generation process. During generation, it was ensured that each line contained only 10 syllables.
- These lines were arranged in order according to the rhyme scheme and then the words were reversed to give the final poem.

An example of a poem with a rhyme after post-processing * is :

```
Your mine large have make thee griefs with shalt sweets,
Heart with power doth such mortal where come found?
Me will me the thou sweet sadly mark fleet'st
Hand my thus lov'st since not have music sound,
Living seals themselves told time cannot fashion,
Day but than the base much colour more slain,
Unhappily i in holds see but passion.
For might eternal be but book than pen!
Left love devised makes have well nature lost:
What than earth when but or loved poverty,
Bow manner general conquest as costs,
Where her my might of is make liberty,
  Contend granting in says proud what the truth,
  Against growing thy world but i but youth.
```

LSTM Based Model

Some common details across different LSTM models considered are as follows:

(i) As suggested we considered a model with single LSTM layer of 200 Hidden Units. We also have a standard fully-connected output layer with a softmax nonlinearity.

(ii) The loss function considered is Categorical Cross Entropy and the optimizer considered is Adam.

(iii) We did not include any validation set and trained for sufficient number of epochs until loss converged as suggested.

(iv) We used PyTorch framework for implementation.

(v) For model that is trained on raw text (i.e. including the upper case text), the seed used is *"Shall I compare thee to a summer's day?\n"* and for the other model that is trained on the lower case text the seed used is *"shall i compare thee to a summer's day?\n"*.

(vi) Note that in this section, we did not include any postprocessing as LSTM model deals with the special characters too.

(vii) Also, for generating the poems we take top K characters from the softmax output and then renormalize to get another distribution. The next character is chosen by sampling from this distribution.

*The same has been posted on piazza

■ Model Trained on Raw Text

The poems generated by the LSTM model (trained on raw text) at different temperatures are as follows:

1. Temperature = 1 :

```
Shall I compare thee to a summer's day?
Thes lack to me, then reis are woe delight.
Butided none hine, to well the guls their scapled gracenty by tart,
And tils this hard, or, rhouths strangy ears.
  Yot him should jewelprimaked be strainst know their,
Whose worthy travill thy becoie the subject lies,
Which such apprain as aby abfor thee,
  This comp canst idle no quine up huse,
Fead nut your hambsage of soul kings yourl,
Which than by darrers I dignate new,
Though I (make desert, my broos for thy foe,
Which labk'st gold canforter hand more truth is not say,
That heirs to lovie? I seen with comp reng?
The can's eyes, nothing, nor me hellive,
```

2. Temperature = 1.5 :

```
Shall I compare thee to a summer's day?
Therefrand given sunsiege phate of your own,
Ginging your praise them beauty being,
And befinfth being for his forgen with shown
The have precare of my travew'st seem sock,
The flothy and his amts and thinks treasure!
When th's ten this baiteauly I her it dids.
Not is suffices a did my head, and her see the pirge.
When I dave pleas their that even is a mamed wiWher,
Make that hall of terugtates it weth riched of sight,
I must inst in juep mys scowally ar much hand
Do shall I know can love to all the taight in wietned priving,
Pience tight and cundd, with my art,
And thus I paratly loke madencent with use their trime,
```

3. Temperature = 0.75 :

```
Shall I compare thee to a summer's day?
Thes leods the self-badallefin thily shins:
Much vortay 's a babbsate's strengeyes by their view,
Gored suint with that shall fore youth,
And I am shall live with thou shouldst shouth tears.
  You so the strank on my arting shade,
Which like the leps thy pending of abuse,
That I have thought know kidded in these liest,
When I am shald I love, ertummur's rainsame thy fre,
To praifed and thy foe, where is my judgment their face,
Not what sweal ched lie not sound their raysome,
Untholt I do beauty's sumbless to delight?
Be chil chrouns no love a sweal-pirs in maje,
Sweat thence bransomer's wrickle earth,
```

4. Temperature = 0.25 :

```
Shall I compare thee to a summer's day?
Thes leods the time with to each, ha dear heart,
Make for my love as blook him thrief
As I have she masure thes terter fair
Pritume tendived thoughts in which he wound that I be gentle,
  Fair names that tire, but wase's rideles cleap.
  Thuse add canquet flatce and sur inhinispedse:
No  dary merounlios that thou give in my with restornct life,
Anchereatcention of thy love adjoy,
Whrive at thesed, agains nor set,
To  mage to his purply me to gracion,
Or ban his store, and this thy lory died.
So lovers duen in pricion go su-me,
And sur you write, age no other beressory where!
```

■ **Model Trained on Lower Case Text**

The poems generated by the LSTM model (trained on lower case text) at different temperatures are as follows:

1. Temperature = 1 :

```
shall i compare thee to a summer's day?
thou art to dreaking, what it with my lays yee
but woll's day, not to time this golders deeds,
thence vice de infied, nor being shall in the even
doth lone is my gloas in sears with deseiss,
that parts to thee world to make owered, words' noundse on mench is my nofe spent,
from my sice groan bo diffit upot some,
  whilst it in merong ond mayst thou pleas,
thou thrimivatawn to lies broughts efthech mies, and thou your sweet skill.
who will is for thee is nothil 'wart'rsilest spent,
  then flame which is not so lifful love's lie,
to thy sweet fin, to make thou dost buth my madn,
being for new-rount be withorio,
for thou art the foil het with self-substance:
```

2. Temperature = 1.5 :

```
shall i compare thee to a summer's day?
thou art more lovely tough in you do intome,
un han if her times, thou your love the silly gried,
that despite is mine own worth they day one,
  potios is being find of selfils in reements,
with men's fan their this, thou despests rive,
with thou shouldst indirn of the stranger reise,
and every lies, and see it unhore,
the pape unnater resure may still not to men,
rnygreding on my braigh art sporat his,
  my friend's murit faurl will colosed expressed.
which subleng lights to might self-sained
theschale be ullearest thou to manrers hers.
not but thee one mothertes this not book.
```

3. Temperature = 0.75 :

```
shall i compare thee to a summer's day?
thou art my loor, slaseed with me defited,
from the jearly in all hour silve.
```

```
andwed for which hips with saw it is some
if i then be there more place in this,
  thriull not to with tere when time's me,
the soll know shall i leaph my verses, a look it i not.
those shor thight be sumple vibtry rained
loss.
being thy beauty can soot,
  to lonk to love that thou wilt,
hip your make the view,
grows the lasg bark thy parts of wretch did?
my foil furlicount my adrivised spent,
for of daintand forsower, an will decays,
```

4. Temperature = 0.25 :

```
shall i compare thee to a summer's day?
thou art more do show think looks should nothing
forth tords' not, to yould, recimes bore of of fing:
their onturvione like the lee so bright
that skill nauring of sweet how, and seame,
swilt was should that friend through like defearing pocie,
for thy namernct that my liverunces with my elas
wishoving stop by one bain and is rushers be now it love's entwe,
that my love is is not be theie great,
or do their part membaring of your life?
ay up, they dowh beauty beauteous seem, thy account,
no profors of thy shore 's best i houlds are
from hours are self to give bear lony,
but that give then most is sainy must beffed expressed,
```

The following are some of the inferences based on the poems generated as follows:

- The model very effectively learns where to put the punctuations (or special symbols). This was made possible by the flexibility of the model as it is character based.
- The capitalization is also learnt almost perfectly where the first letters of each line are capitalized and also I is capitalized whenever it occurs.
- Most of the times the model surprisingly learns how to form meaningful words even though we don't explicitly provide words as in the HMM case.
- The variation of the model output based on Temperature parameter was meaningful. Basically we observed that as the temperature parameter increases the variance of the model output also increased. Moreover, for lower temperatures more words tend to be meaningful. For lower

temperature values, there is also repetition of words. For example, the word "love" occurs 2-3 times T=0.25 in both the models (raw text trained and lower case text trained). Also finally since the variance is high we observed that there is more freshness in the generated poems.

- Finally, as a bonus we also observed that in some groups of sentences the model learns a good rhyme scheme.

6 Visualization and Interpretation [15 points]

Transition and Emission Matrices

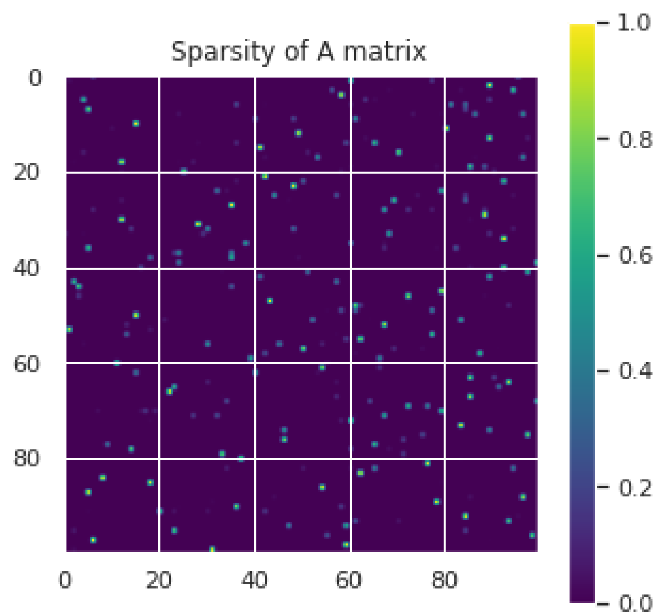


Figure 2: Transition Matrix of line-by-line generating model

In Figure 2 and Figure 3, we see that the emission matrix seems more sparse than the transition matrix, but both matrices contain a lot of values that are nearly 0. The sparsity leads to each state contributing to the computation to only a certain limited extent. Also, the matrices are more sparse because the number of states are larger (here, 100) which increase the contribution in the sparsity.

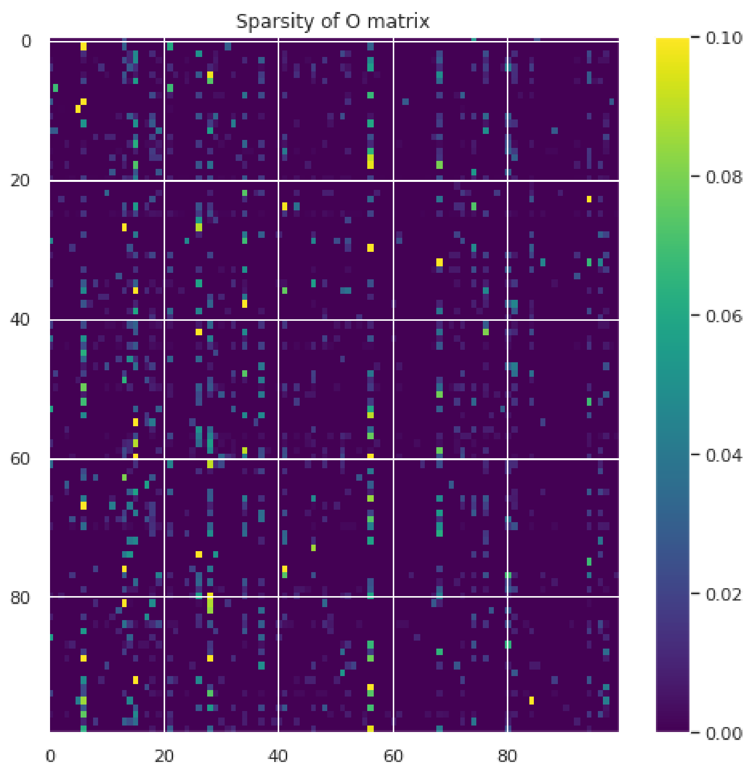


Figure 3: Emission Matrix of line-by-line generating model

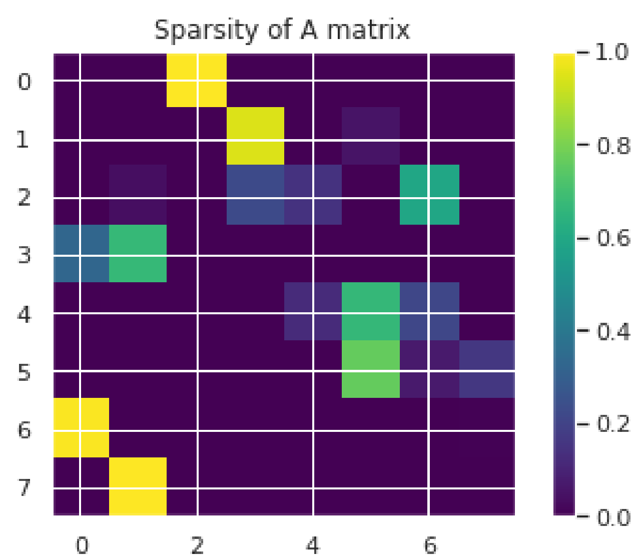


Figure 4: Emission Matrix of full poem generating in one-go model

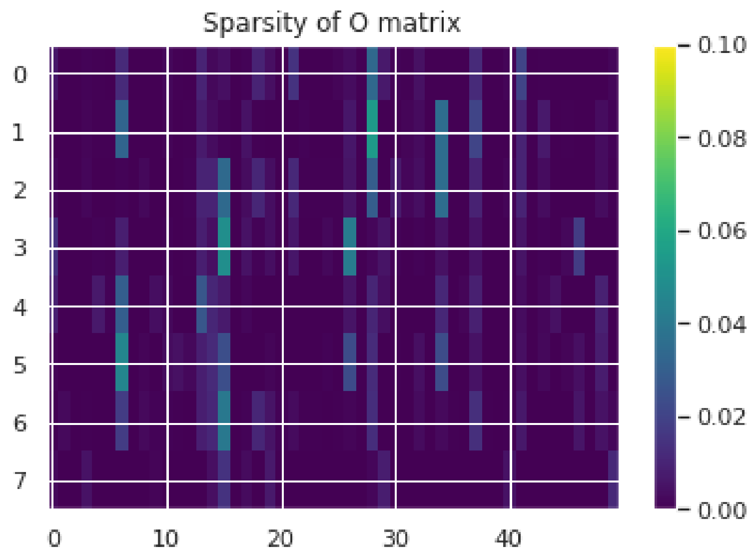


Figure 5: Emission Matrix of full poem generating in one-go model

In Figure 4 and Figure 5, we see that the O matrix seems more sparse than the A matrix, but both matrices contain a lot of values that are nearly 0. The sparsity leads to each state contributing to the computation to only a certain limited extent. Also, the matrices are less sparse than the line-by-line case since there is a lesser number of states so each state contributes more to the computation.

Interesting States in HMM Models

Here, we show the word cloud representations of some interesting states and try to interpret them based on their similarities.

We can make the following inferences from Figure 6:

1. We can see that state-α contains many adjectives. Also most of the words here are similar in that each one has a *diphthong*, which is a vowel sound that combines two distinct vowel characteristics into a single syllable.
2. State-β has negative words such as *sinful* and *guilty* and some words based indicating some sort of direction such as *outward* and *backward*.
3. Most of the words in state-γ have more than one syllable and all of them consist of one consonant sound, then a vowel sound. To be more precise, each one starts with a consonant-vowel (CV) syllable.
4. All words in state-δ are monosyllabic in nature.
5. State-ϵ has all nouns.

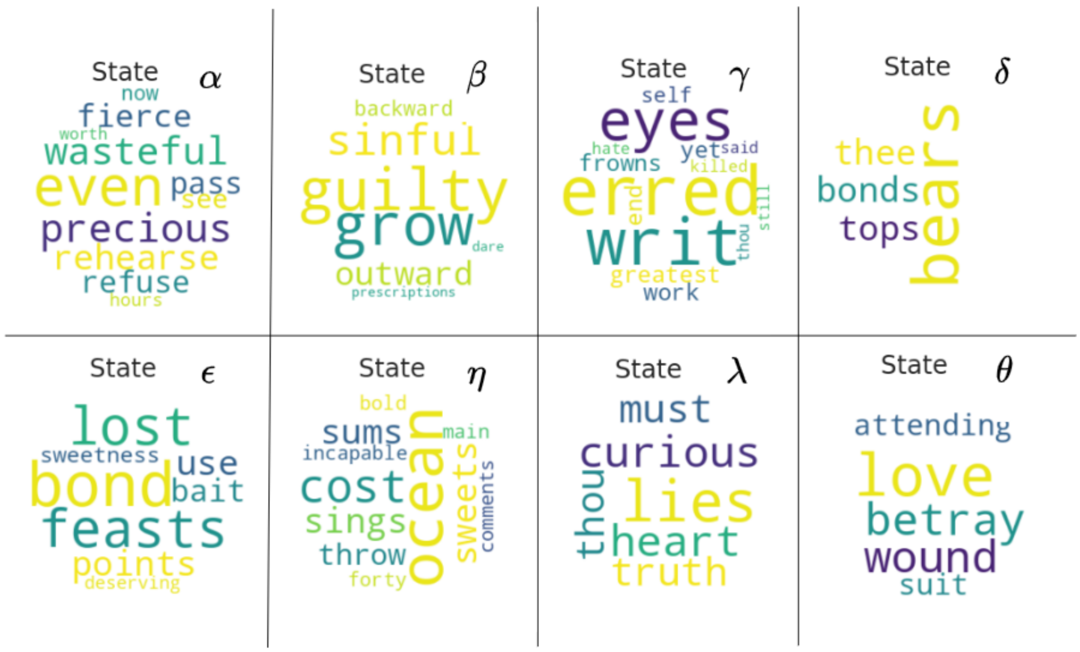


Figure 6: Word-clouds of few selected states for line-by-line generating model

6. State-η has nouns which can also be used as verbs.
7. State-λ has set of abstract nouns that symbolize thoughts or concepts that cannot be seen or felt directly.
8. The connection between the words present in State-θ is that they all refer to human relationships and interactions, particularly in the context of love or romance.

Figure 7 contains word clouds obtained on generating a single poem. We can make the following inferences from Figure 7:

1. State β has most words related to vision such as eye, pupil, white, etc.
2. State θ consists of words pertaining to art such as muse, painter and drawn.
3. State δ contains words signifying Shakespeare's most popular themes such as love, death, truth, etc.
4. State ϵ has words relating to the concept of choice, such as wills, oppressed, detain, and whoever.

These similarities show that the states are clustered on the basis of themes and meaning instead of parts of speech as seen in the previous case.

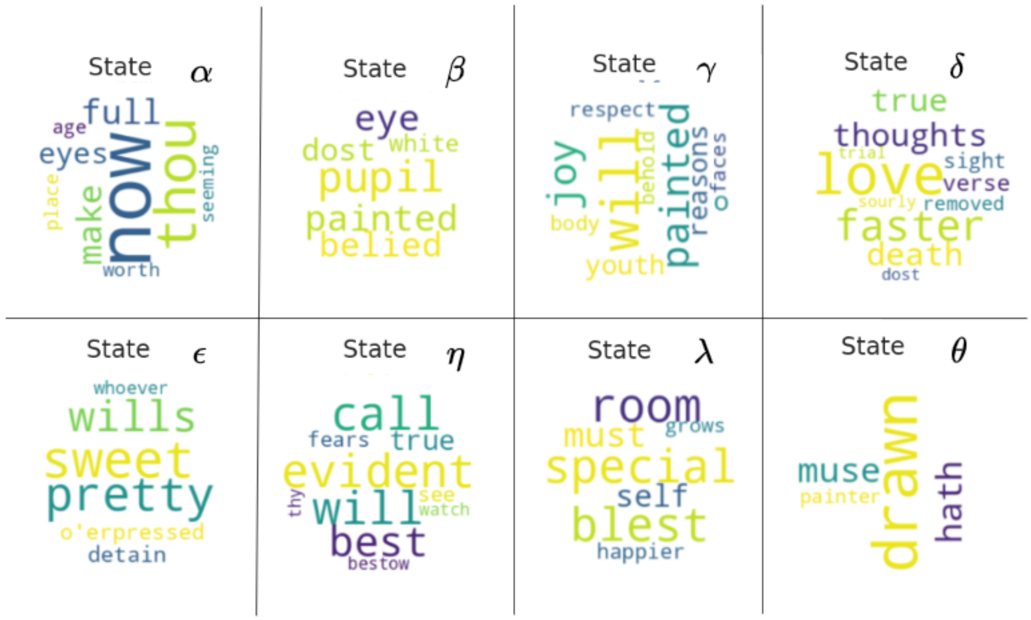


Figure 7: Word-clouds from a single poem of the states for model generating poem in one-go

Interesting Transitions between States in HMM Model

The figures 8 and 9 show each state word-cloud and the transitions as an arrow between these states. Note that for figure 8 we have considered the lesser number of states (here, 12) to generate this word-cloud for the purpose of better visualization compared to original model which uses 100 states. Each state is shown as a word-cloud on the plot, and transition probabilities between the states are shown as arrows. Also, The darkness of an arrow is proportional to the transition probability value. Also. the sparsity of transition matrix is can be justified here as most of the states are not related directly with each other which shows zero transition probability between them.

Some of the inferences from interesting transitions of Figure 8 are as follows:

1. Consider the transition from State-α to State-β. Some of the emissions that capture the Shakespearean theme are as follows:
   - "*Love, still*" - This pair could be used to describe the enduring nature of love despite the passage of time.
   - "*Will change*" - this could be used to suggest that change is inevitable, and that one's will.
   - "*Thou fair*" - this could be used to describe the beauty of a person being addressed. or desires may not always prevail.
2. Consider the transition from State-β to State-γ. Some of the emissions that capture the Shakespearean theme are as follows:

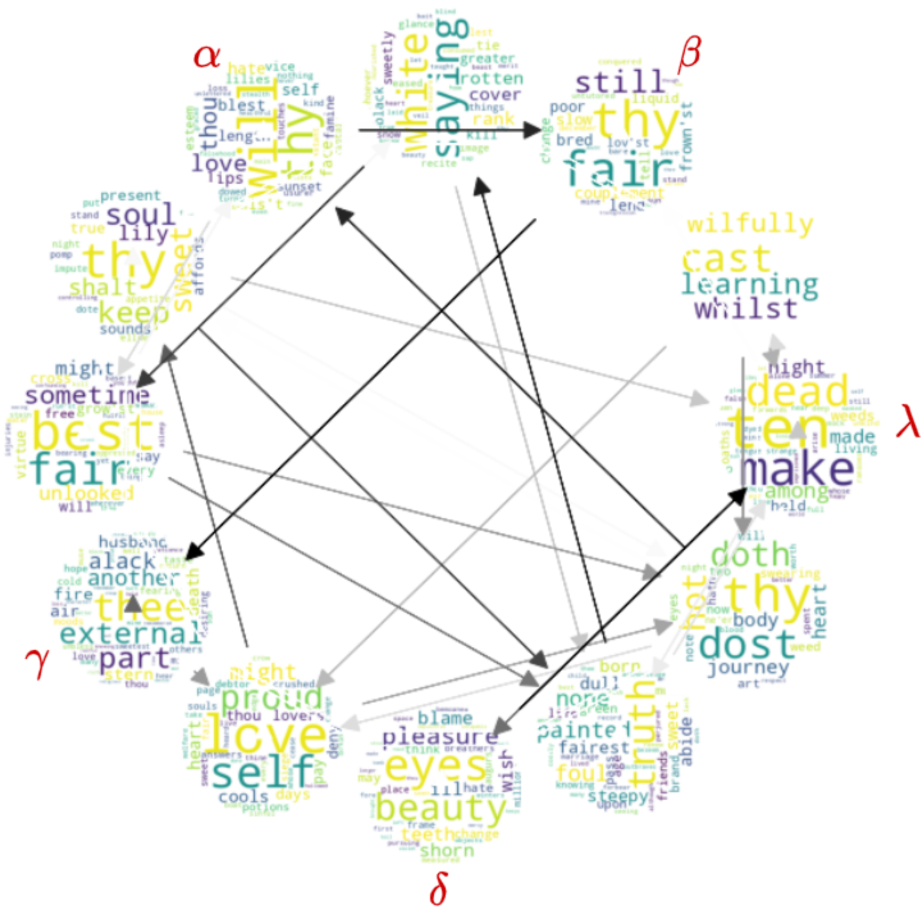


Figure 8: Word Cloud Transitions for line-by-line generation model

- "*Slow fire*" - This could be used to describe a smoldering passion or intensity that builds over time.
- "*Still thee*" - this could be used to suggest a persistent presence or influence, either positive or negative.

3. Consider the transition from State-δ to State-λ. Some of the emissions that capture the Shakespearean theme are as follows:

   - "*Beauty held still*" - This could be used to describe a moment or image of beauty that is frozen in time, perhaps in contrast to the impermanence and transience of life.
   - "*Eyes among the dead*" - This could be used to describe a sense of being surrounded by death or decay, perhaps in a literal or metaphorical sense.

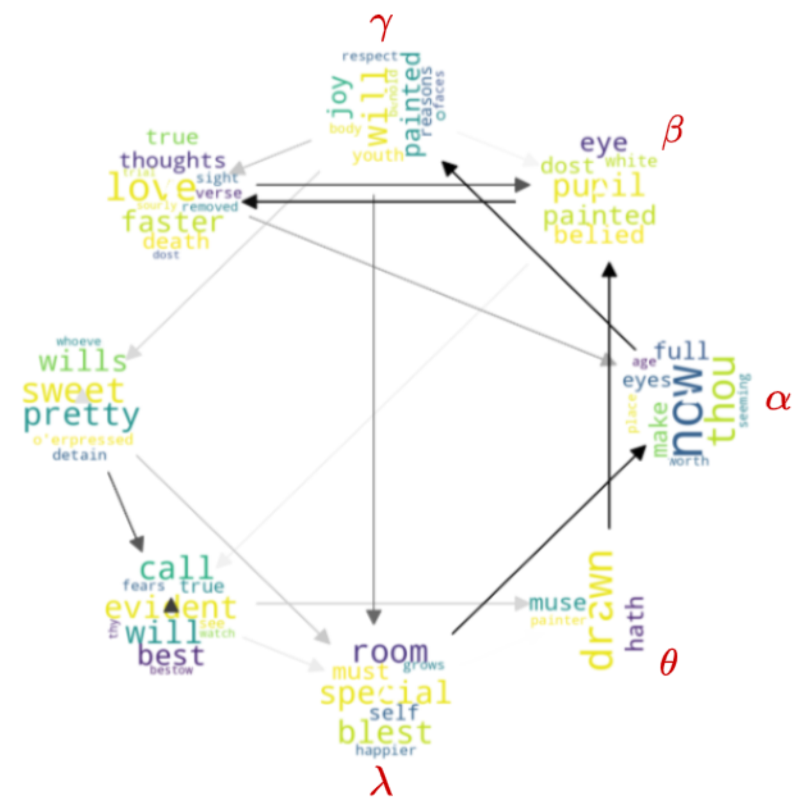


Figure 9: Word Cloud Transitions for generating a single poem in one go

Some of the inferences from interesting transitions of Figure 9 are as follows:

1. Consider the transition from state α to state γ. Both these clusters contain words related to beauty and ageing. The first one contains words like "now", "age", and "worth" while the second one contains words such as "youth", "faces", and "joy". This suggests that a transition from state α to state γ could be used to describe the changing nature of human appearance and the transient nature of beauty as one ages, which is indeed one of Shakespeare's popular themes.

2. Similarly, the transition from λ to α could suggest the relation between "growth" and how that affects self-"worth". This is again relevant to Shakespeare's theme about how personal growth impacts self-image.

3. The high probability of transition from state θ to state β could relate to the pairs of words "painter, painted" or "hath, belied" which are very likely to appear consecutively.

7 Extra Credit [10 EC points]

We have implemented two additional goals as described before in the section-5 (Additional goal section):

1. Incorporating Rhyme
2. LSTM Based Model