1 Introduction

Our group is composed of Jason Yang, Pierre Walker, James Wang, and Cyrus Fiori. Our team name is *The Navier Folks*.

Tasks were divided as follows:

- Jason Yang: Implemented RNN and wrote up relevant sections.
- Pierre Walker: Implemented HMM and completed Additional Goals.
- James Wang: Implemented HMM and completed Visualization & Illustration.
- Cyrus Fiori: Wrote up the HMM sections, formatted and edited report.

We used the following packages: numpy, matplotlib, pandas, and PyTorch.
Link to Colab: Code
Link to Piazza post

2 Data Pre-processing

Hidden Markov Model

Training was done on each line instead of the entire poem. This was done in an attempt to preserve the characteristic iambic pentameter structure of each line, which would capture Shakespeare's voice. The drawback is that this training method did not capture the relationship between lines. We also removed punctuation marks from the raw data, with the exception of apostrophes found within words, in an attempt to avoid nonsensical punctuation insertions in the output poem. In order to define what counts as a word, we referred to the given Syllable_dictionary.txt file. Capitalization was not removed, with the intent of preserving capital letters in words at the beginning of sentences and of pronouns (i.e., I). This was done at the risk of our HMM randomly capitalizing or switching to lowercase throughout the poem. Hyphenated words were treated as two words plus a hyphen to allow for flexibility in generating hyphenated words in the style of Shakespeare. However, this increased the risk of producing one or both words in the hyphenated compound as gibberish during poem generation.

Recurrent Neural Network

Please see the recurrent neural network section for a thorough description.

3 Unsupervised Learning

In this project, we decided to re-use the Baum-Welch algorithm developed for set 6. Our code followed the subsequent steps:

1. Random initialisation: We initialise the observation and transmission matrices randomly.
2. E-step: We compute the marginal probabilities using the Forward-Backward algorithms.
3. M-step: We update the model parameters based on the maximum likelihood estimate given by:

$$\arg\max \prod_i P(\mathbf{x}_i) = \arg\max \prod_i \sum_{\mathbf{y}} P(\mathbf{x}_i, \mathbf{y}) \tag{1}$$

4. Repeat the E and M-steps for 1000 iterations or until the model parameters have converged.

To determine the optimal number of hidden states, we trained multiple models with incrementing number of hidden states (1, 2, 4, 8, 16, 24). From this, the only poems which made any sense were those with 24 hidden states; as such, we decided to use this number of hidden states.

4 Hidden Markov Model

The sonnets generated using the HMM are displayed below.

Hidden states = 1
Tyrants been not more me a the of the
Bear shadows make bears able for be those
Wand'ring subtleties excuse shouldst faults and
Thou to cold in I do not ill help mine
I thou follow filled scope so she of and
Proved the know prove care to sweetest fever
World spent made truth doth muse forgot who from
Bonds time we of of will fair need true an
Hid be bath more boldness then worthless the
Therefore kissing old I as above do
All win being a after-loss are count you
And eyed I I book less pale crowned best
Brave and thou says constancy untrue soul
See thy being such in wrongs took young if

Hidden states = 2
The which holy been my in dost not black
There more the for that but my touches the
He at by is mended sight figured but
Heart of my thou count ah you ay leisure
Hours more of night good seen pale my and that
Thou show thou attaint thing mine in voice what
Enlarged art in proof say live thoughts hath blood
To deceivest self thou fortune shade best
Left and which increase chide fly spirit wife
Forbid is and debate contain my he
Tables looks with slandered is that and fashion
Maiden my young not more that that here's heaven
Confess seen my lose more end the I thy
Of as dead not yet strong foul hang that form

Hidden states = 4
Familiar argument it turns ill be
Time yet or excellence thy have or fast
Thy give the faults be such in then times all
Nor of which me on beauty's night in edge
Thy in is thought should thou mine of men ye
Hand thee to sleep winter my stain soul lies
Spirit men and tongues the sacred by place but
To that store rent for and do will tell doth
Yet reason for I purpose upon thee

But well thinking then days not straying I
Not drooping to worths thou then but thy poor
Own shall nor make princes to who though those
Looking nimble loves frank no death to but
Writ secret me of a dispraise the world's

Hidden states = 8
Dear pen two comfort wastes and many night
It the true on thou at love directly
Some thy mind rare basest great welcome by
Crime with that to of I all what rose thy
Love with my I creature the vassal his
Believe invent treasure please praise being pleasure
Others I hearts my nightly cross than greater
Bright the nought with with survive me for verse
Is the her was that a have O they men
To two art from and but red figures the
But vile time his broke much falsely husband
Be heaven's thy bower worthy day continual
Show wealth to and their war be it flattery
Confounds and loss my breasts what found wood what

Hidden states = 16
Minion the esteemed pen there my steeled teach
Novel their crests poisoned if earth crowned
Riches face on to partake my not deep
Receives is I doth my is to engraft
Were of I this expressing do heart may
Nor tongue who sweet like neigh that well hand are
Of will when it up thee be sweet of my
Live that make live subject but to your breast
Me of fair be of thy hot soil thy heart
Deeds I ah it weak trust of you love in
Sin absent perhaps sweet dedicated
Half longer may found numbers to teach loved
Possessed but the their love thievish lovely
Days elsewhere affairs woman's kind-hearted

Hidden states = 24
Feeling worthiness short if my fears as
What nature's scope is life's life as thee as
Lively day to belong some cheeks of my

Should I thou in rhyme aspect with his and
To so black disdain of win and heart his
Then at thy friend arts image that be spent
Cruel gilded that I bars time works mine
It and this that life to fits the thoughts state
Tongue of rehearse the eyes when interest self
Thousand they all sense but the count how by
Predict now yet have full on haply no
My sweet is the physician is more child
Mistress behind in no whose of beside
Lovely true of in sessions and nothing

The created poems mimicked the atmosphere, syllable count, rhyme scheme, and length of Shakespearean sonnets. They retained Shakespeare's original voice to an extent, capitulating an airy, mysterious, and grammatically ambiguous tone characteristic of Shakespearean poems. It appears that the HMM was able to capture the poetic voice of Shakespeare, likely by learning the relationships between words from the original poems, and successfully applying those rules to formulate poetic sentences in most cases.

Despite capturing the voice of Shakespeare, these poems were noticeably different from Shakespeare's own work as they lacked a certain refinement. Notably, some grammatical errors clearly indicated that these poems were computer generated by a model which had not mastered the English language to the same extent as Shakespeare. These included repeated words (i.e., ”... with with ...”), nonsensical sequences (i.e., ”Cruel gilded that I bars time works mine”), applying words incorrectly relative to their grammatical function (i.e., ”Minion” used as a verb instead of a noun), and nonsensical words (i.e., ”attaint”).

When examining how the number of hidden states affected poem generation, we observe that at low numbers of hidden states (i.e., 1), the sentences are less coherent, with nonsensical repeated words (i.e., ”I I”), and ending on conjugations (i.e., ”... if”). This makes sense as the model would be fitting the data more coarsely without the opportunity to fine tune its parameters. As the number of hidden states increased, the poems generated became more coherent and certainly more poetic (i.e., 16 hidden states: ”Sin absent perhaps sweet dedicated”). This is congruent with trends in HMM, where as the number of hidden states increases, so does the ability of the model to fit the data, by making more passes through to further reduce the training error.

5 Recurrent Neural Network

link to code
We implemented our recurrent neural network using the Pytorch package. Our architecture uses a single LSTM layer with 200 hidden units. Afterward, the last unit in the LSTM is passed through a dense feedforward layer. Our model takes as input 40 characters and predicts the next character, by minimizing categorical cross-entropy loss after applying a softmax to the output layer. For the inputs, the characters are one-hot encoded by 65 standard characters, which encompass the entirity of the Shakespeare corpus.

The only preprocessing we do is eliminate the numbers at the beginning of each sonnet. We choose to leave all of the "\n" characters so that the RNN can learn the ends of lines and the ends of sonnets. We then split the entire Shakespeare corpus into 97634 sequences of 40 characters, with an offset of 1 between each sequence.

Our training set consists of this set of sequences of length 40. While we also tried using smaller training sets by increasing the offset (using semi-redundant sequences), we did not find that the generated sequences were as good. We fixed the learning rate to 1e-3, which seemed to stably decrease the training error with each epoch. We tuned the number of epochs for training and found that 150 epochs resulted in good sequences. Even though the training error did not converge fully, we stopped training earlier to avoid overfitting. While, we also tried predicting more than just the single next character from the RNN, we found that the generated poems were less realistic.

For poem generation, we started with the seed sequence and generated characters one at a time based on the 40 previous characters. For non-zero temperatures, we divided the logits from the output of the RNN by the temperature before applying a softmax and then sampling the next character based on these probabilities. For zero temperature, we simply used the most likely next character as the next character in the sequence.

We find that our LSTM successfully learns sequence and sonnet structure. For low temperatures, most of the sequences follow iambic pentameter with the correct number of syllables and flow. Similarly, these lines end with some sort of punctuation, followed by a new line, which is the correct structure for most sonnets. Each new line begins with a capital letter, which is also realistic. Unfortunately, it is difficult for the LSTM to pick up on rhyming structure, as most lines in the sonnet are 40 or characters, which means that the rhyme in the line before is not considered in memory.

Overall, the LSTM seems to generate poems that have the correct vibe for a Shakespearean sonnet, compared to the HMMs which do not seem to flow as well. Furthermore, the RNN is more aware of the parts of speech that follow each other. However, the training cost for the RNNs is higher and it seems to require more data to learn successfully on a character-based model. We were able to significantly speed up the training of our RNN by GPU acceleration on a Nvidia Titan V, so our models trained on the order of minutes. However, if we trained the RNN on a single CPU core, like the HMM, it would take significantly longer.

Increasing the temperature increases the variance of the generated poems. While the poem generated by a temperature of zero is rather conservative and repeats the words that Shakespeare uses most often, a temperature of 1.5 leads to much more variety but also words that cannot be pronounced in the English language, and significant breaks to the traditional format of a shakespearean sonnet. It seems like a temeprature between 0-0.75 generates the best balance of interesting and realistic poems.

Example poems from the RNN:

Temperature = 0:
shall i compare thee to a summer's day?

Thou art more loves to the very abrous fidle,
And yet not still even his black with mune.
In thy sweet formering of their world expues,
For that say more that which fline in seem,
And say in his beauty shall to thee doth live,
And since honour more than thou souldst in me,
Hine eyes, your lous'st me not so much rest
Onfeiticy, but by thy self with pure han eres.
Then stoply fair that my time that case sone.
To tiully me to make our name, she less
Is wond of mine on shamonous a many.
Naturay drien presage of thee despised,
But

Temperature = 0.25:
shall i compare thee to a summer's day?
Thou art me, thou should thou wilt besmoun,
As those lipse that this shall gave my heart'st burt,
And so my plays where my love as your lies,
And so by these vainted acter is bend,
That for this thought goner, which is so greater,
But not then thou my love tame, her the sun
outs do turn for my says our presict triff
With leathe her to thee requite upon brow?
Why of those would beauge we lack upon thethted,
Mine own truth voights which to ensuray peeds.
You whom she is near suity to steep to be.
Af of hunot after that I in thy art?
Or cally of poor and

Temperature: 0.75:
shall i compare thee to a summer's day?
Thou art more bonds in their murity sight,
Ditied by hister-blace eachers wretch doth lack,
Whon of love that terges to every hade,
So those that which thou restaling so vend.
If thou most love to whose rendos excellegħed:
A trave alter in all alone shall tone's past.
Letthenoud, bluck intermita inds dow:
To sourkent thou the least, some false old,
And keeps thee I have still the store,
But sweets of thee, 'Nat this auris affooted.

I ne' from me for dowh hell with d-ain heece,
And by that I my body the contenced,
And but a fil

Temperature: 1.5:
shall i compare thee to a summer's day?
How have eye knoulgly detyoly chadly lien,
To play as for hore, why when that you should po most can,
He capenury cantel fronds be recods,
Or can uper canngry in other bold,
Thy objeess wat as your sweet as doth,
Unter theme leach unless to sleep, there graines.
How makenemant, more int'ring with mancest,
The clear besteet forly sleak extile hey,
Goor better it were beremone) shall wond colfaded note.
Orieviol, whoch welldingums devosed view,
Yet I this, manying no, and my love swear,
Thy should kind, but they with thy altaints s

6 Additional Goals

We decided to incorporate a rhyming scheme in our project; specifically, sonnets (ABAB CDCD EFEF GG scheme). This is indeed something that is missing from our original implementation of the HMM model.

Before we are even able to think about generating rhyming pairs, we must first process all Shakespears' sonnets to identify rhyming pairs (this is straightforward as they all follow the same rhyming scheme, save two). With these pairs concantenated in a single dictionary (rhyme_dict), it was then possible to generate the sonnets.

To generate a poem with a rhyming scheme, we need to generate it stanza by stanza where, within each stanza, two rhyming pairs had to be generated using a pair of words from within our rhyme dictionary. Once the pairs of words were selected (the sentence seeded), we then needed to construct the sentences in reverse. The difficulty with this, however, is we are starting our sequence generation with an observation, rather than a state (as the original case). This proved to be a challenge; however, from Baye's rule, we know:

$$P(\mathbf{y}|\mathbf{x}) = \frac{P(\mathbf{y})}{P(\mathbf{x})} P(\mathbf{x}|\mathbf{y}) \tag{2}$$

where we can assume $P(\mathbf{y})$ is uniformly distributed and $P(\mathbf{x})$ is a constant, allowing us to compute $P(\mathbf{y}|\mathbf{x})$ (after normalising to 1). The states can then be sampled at the given probability to give us the most likely state to generate this observation. From here, the rest of sentence can then be generated (after reversing the sentence once generated). Below is an example sonnet (with 24 hidden states):

Yet of the therefore that it at other

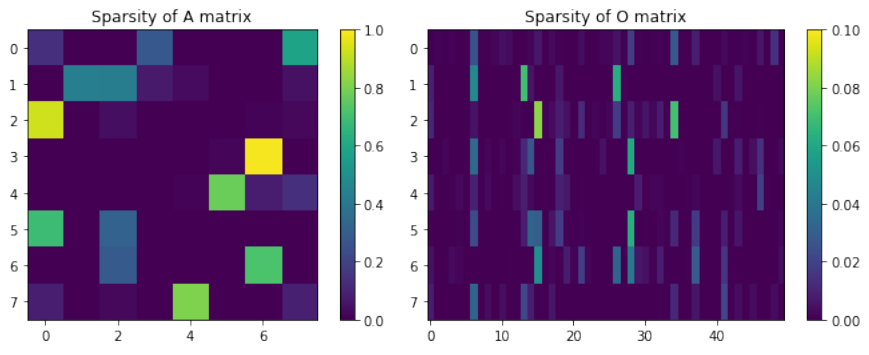

Figure 1: Figure 1: Sparsity of transition and emission matrices.

Vainly with then metre times of the doubt
Two array lack as swift winter smother
Presence when to shalt set might be about
Fashion to to his and world's waste rage to had
Spirit in by youth vaunt of constant reason
Never me when of heart to will set faith mad
It full suns gav'st even ah are treason
Shames but earth the in for he night of cause
Now did find despite but if are am mad
It that I cannot take in thee will laws
It that and of and virtue youthful bad
Falls of them sunset no rotten defaced
Being see their receive I false the down-rased

The above has clearly satisfied our requirements for a sonnet, now matching the rhyming scheme. However, it still doesn't quite match what our expectations are for a Shakespearean sonnet; if anything, it appears a bit less Shakespearean. A slightly more flexible implementation, rather than just relying on the rhyming pairs available in each sonnet, would be to classify words by their ending syllable and form a more-extensive rhyme dictionary. We may have been able to generate a more-reasonable sonnet by reducing the constraint on the rhyming pairs available.

7 Visualization & Interpretation

As an example of the sonnets generated by our Hidden Markov Model, We first visualized the sparsity of the observed and transition matrices, shown below:

Evidently, state 3 is predicted to follow state 6. Similarly, state 0 is highly probable to transition to state 2 according to our model. Similar correlations between states have are less strong, but in general the transition matrix is sparse i.e. most cells have values close to 0. This indicates that state transition are well-defined:

You but that he i spent nor

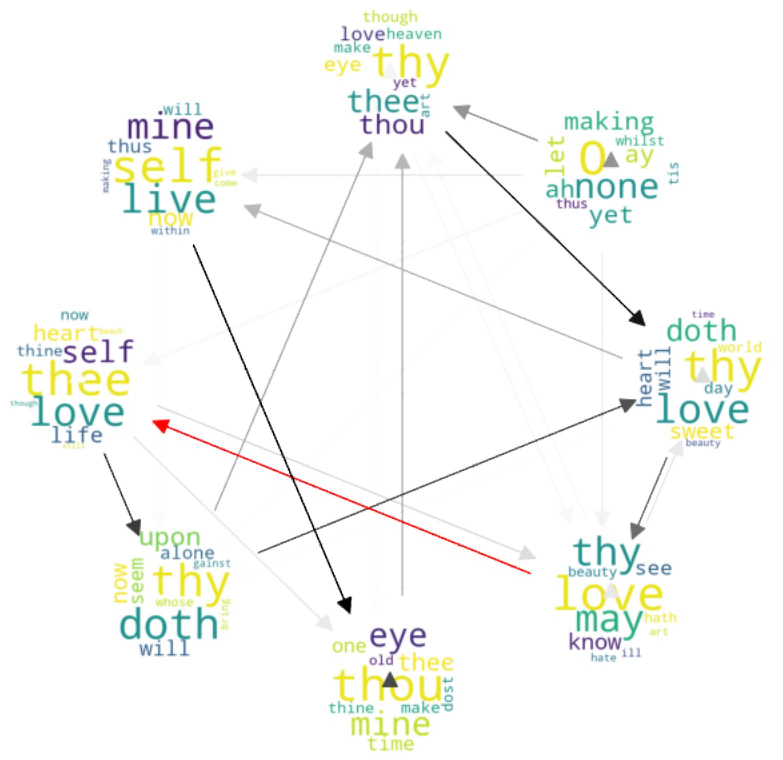

Figure 2: Wordcloud and transition probabilities among HMM states

from one state there are only a few states that are likely for our model to predict.

Our observation matrix also looks quite sparse, indicating that most words tend to associate with only one or two hidden states. However, there are words that associate with many hidden states, like word 8. We visualize the top words 10 words in each state in wordclouds below:

We see the trends we observe by looking at the transition matrix: some state transitions (e.g. first state (0) to third state (2) from top, clockwise) are highly probable while most other transitions are not. Looking at the wordclouds, we can see that most wordclouds are organized based on parts of speech (state 0 consists of nouns and pronouns, for example).

Strong transitions between states capture features of English syntax. For example, from state 0 to state 2, the transition probability is very strong. Since state 0 consists of mostly pronouns and nouns, and state.2 consists of many verbs, this transition probability makes sense, as in English verbs often follow directly after the subject of a sentence.

Hidden Markov Models learn patterns by first guessing emission probabilities for all states and transition probabilities among those states. It then attempts to group words with similar function/meaning in the same state while also learning the relationships between states using the Baum-Welch algorithm. The Baum-Welch algorithm calculates observation and transition probabilities repeatedly sampling and adjusting probabilities until the learned transitions/emissions agree with training data.

Finally, we attempt to assign meanings to each state. Starting from the top wordcloud in Figure 2 and proceeding clockwise,

- State 0: pronouns/nouns often used as subjects
- State 1: articles, adverbs, and conjunctions often found at beginnings of sentences
- State 2: verbs
- State 3: nouns often adressed as objects (as opposed as subjects)
- State 4: nouns that can be used as either subjects or objects
- State 5: Prepositions and helper verbs
- State 6: nouns relating to relationships
- State 7: nouns relating to the self

8 Extra Credit

Not attempted.