Introduction

The problem that we are working with is generating new text based on an author's book or piece of writing. While I can't think of an enterprise reason this could be important, it definitely is super interesting to me because if we fine tune a model sufficiently, we could have it output writing that can be near indistinguishable from an author's actual writing. Essentially what we're doing with this project is, feeding a model an entire book and training it on that book, then asking it to make sentences based on what we fed into it.

Analysis

For this data, all it was is just a full copy of the play by William Shakespeare, Romeo and Juliet. The first thing I did after downloading this from Project Gutenberg was delete everything from the txt file that wasn't the play itself because the file had a lot of terms of service type stuff and legal jargon. The next part and more important part of cleaning that I did was tokenizing and one-hot encoding. The reason this is necessary is because the model cannot accept anything except for numbers, so we must turn the words of the play into numbers that correlate to their words. The first step of this is tokenizing. To tokenize the book we just divide it word by word. The second part of the cleaning is one-hot encoding. As mentioned before the model can only work with numbers, and one-hot encoding does exactly this. What we are doing when we one-hot encode is assigning a unique set of numbers to every individual word. This allows every unique word within the play to be represented by a vector of numbers and allows it to be used by our model to learn.

Methods

For both models, I used the LSTM architecture. I decided to use this architecture for both models because it is a recurrent neural network that functions best for this type of problem because it uses past states, along with the current data to make predictions. This works especially well with this type of problem because a book, or in this case play follows a sequential order first in terms of content and in terms of patterns of words and I thought a LSTM would be best. For the first model I used a simpler model architecture with just the one recurrent layer and with a dropout layer, where all points have a 20% chance to be switched to 0 in order to prevent overfitting. With the second model I used a deep LSTM with multiple recurrent layers. What makes this different from the prior model is that the full outputs of the prior layer are used in training the current layer, as opposed to just the output of the last time step. This makes this makes the second neural network much more interconnected and further relying on other parts of the data, in this case the play, in order to make itself more accurate

Results

My deep LSTM performed a good amount better than the standard one. The deep one had accuracy of .71, while mine had accuracy of .46. This is a pretty big jump when we use more

layers and make them fully recurrent and it shows in the sentences both models generated. First will be the sentences my model generated, followed by the sentences the deep model generated. Normal model:

Seed text: send to romeo but when i came some minute ere the time of her awaking here untimely lay the noble paris
GENERATED TEXT: come i prince servant romeo romeo romeo madam that thou death not that the montague the may so nurse groan

Seed text: and his brother valentine mine uncle capulet his wife and daughters my fair niece rosaline and livia signior valentio and his
GENERATED TEXT: cousin and daughters of his moved and the lively helena of the unseen button of the watery flask of duellist

Seed text: of substance as the air and more inconstant than the wind who woos even now the frozen bosom of the north
GENERATED TEXT: of the capulets of the capulets sampson the the capulets place and immortal passado of the capulets came the immortal

Seed text: her silver why why with her silver what say you simon catling first musician marry sir because silver hath a sweet
GENERATED TEXT: sound the then the me that both one that a comfort i tell i will for civil within to heads

Seed text: rode i think he told me paris should have married juliet said he not so or did i dream it so
GENERATED TEXT: i am fool of my dear juliet nurse is is house so young bachelor and tell r and tell r

Seed text: pity you at odds so long but now my lord what say you to my suit capulet but saying what i
GENERATED TEXT: have said you thee i am so you wilt so of bid my great romeo i must wed to you

Seed text: thy hand late farewell good night romeo but that a joy past joy calls out on me it were a grief

GENERATED TEXT: some that sadness and would i world my dreams and know and friend it is it pardon you not not

Seed text: here in this city visiting the sick and finding him the searchers of the town suspecting that we both were in
GENERATED TEXT: a house i montague the montague be trust ah the montague i make to die nobleman to have to fortunes

Seed text: them above a common bound romeo i am too sore enpierced with his shaft to soar with his light feathers and
GENERATED TEXT: be bound i be moody and as soon moody and his soon moody and his stirs of utruvio of his

Seed text: a surgeon exit page romeo courage man the hurt cannot be much mercutio no not so deep as a well nor
GENERATED TEXT: a plague my my course nurse lawrence aside to slave to broad to cannot it a it and some the

Deep model:
Seed text: lawrence now must i to the monument alone within this three hours will fair juliet wake she will beshrew me much
GENERATED TEXT: the romeo my night o was be of a be that a grief that a which love as what is

Seed text: men either withdraw unto some private place and reason coldly of your grievances or else depart here all eyes gaze on
GENERATED TEXT: us and to peace and balthasar of death thou in can word of young me tell be tell be tell

Seed text: doth give nor aught so good but from that fair use revolts from true birth stumbling on abuse virtue itself turns
GENERATED TEXT: vice vice vice the vice of the beast of the house and bakes of the watery beams of the watery

Seed text: it to my face paris thy face is mine and thou hast it juliet it may be so for it is
GENERATED TEXT: not mine own you life you that my lady o thou her thou my her love your that man price

Seed text: is paris have i thought long to see this face and doth it give me such a sight as this lady
GENERATED TEXT: capulet unhappy wretched hateful day most miserable in many in the must for a his now and will be all

Seed text: you think fit to furnish me tomorrow lady capulet no not till thursday there is time enough capulet go nurse go
GENERATED TEXT: with her to church tomorrow the think o what the am i the cease of her i her my dear

Seed text: apt to quarrel as thou art any man should buy the fee simple of my life for an hour and a
GENERATED TEXT: quarter mercutio him what be nurse and what juliet still to tender that his my soul my his wisdom him

Seed text: patience your looks are pale and wild and do import some misadventure romeo tush thou art leave me and do the

GENERATED TEXT: thing i bid with for thou world to thou let me do my love with more she and tell thee

Seed text: not what to say peter o i cry you mercy you are the singer i will say for you it is
GENERATED TEXT: is her a some your she you is thus that life that a grave juliet i will i love so

Seed text: of love this unbound lover to beautify him only lacks a cover the fish lives in the sea and much pride
GENERATED TEXT: of the project web of the watery beams of the watery beams to what is is it and goose i

Overall, the text that both models are outputting are somewhat incoherent, but there are some stark differences after further observation that prove that the .71 accuracy vs .46 was significant. The less accurate model has the widespread problem through all 10 generated sentences of having continuous repeated words that make it further incomprehensible, while the model that is more accurate still has this problem, but on a much smaller scale. Additionally, a few of the sentences generated by the more accurate model are very close to being full coherent sentences, while none of the ones generated by the less accurate model are (in my opinion). Another aspect the more accurate model does better than the less accurate one is the use of names.

Although the more accurate model doesn't do it completely correctly either, it seems it was able to learn some of the grammar rules regarding names and use them more correctly than the other model. These pretty major differences really show the power of using a deep recurrent network because the differences were not just numerical, we are able to actually see them, and I bet if I could make a model with even better metrics, I'd be able to generate text that sounds even closer to Shakespeare and maybe even have the model pick up on the iambic pentameter and have it be a feature that is then present in every instance of generated text.

Reflection

The main thing I learned in this assignment was how important multiple recurrent layers are in the use of a RNN. When we first learned them, I didn't see using multiple layers as super important since we already get the hidden state of the last layer with our current data but when using the full output along with the current layer's data I was able to see the predictive power of a RNN with multiple recurrent layers. A struggle I had with this assignment was preprocessing incorrectly. First I used preprocessing similar to what we'd done inc classwork except using words instead of characters, and then I did another step of preprocessing in order to make sequences just like it was shown on the Jupyter notebook given and doing it twice caused for some really wonky errors that led to me needing to take extra time on this. What I learned from it

was that typically, the most simple solution is usually the best one and in this case the solution was less computationally intensive as well.