Background: Shakespeare is one of the most influential and widely studied authors of the English language. Over his career, the author has more than 35 plays attributed to his name. However, there is not a definitive number of plays that he wrote in his lifetime, as there are some works that he is suspected, but not confirmed, of writing. One of these plays, a history called Edward III, was published under an anonymous name in 1596. Many scholars suspect that Shakespeare had a hand in writing the play, although the consensus is that he is not the sole author. As such, different scenes are suspected of being written by other poets of the time, most notably Thomas Kyd.

There have been previous machine learning publications that deal with Shakespeare's folio, although at the time of writing this paper, Edward III has yet to be the target of these methods. These projects range from classifying entire works as written by Shakespeare to focusing on a single play, like Henry VII (which is another "anonymous" play), which was done by a lab at MIT and shown below.

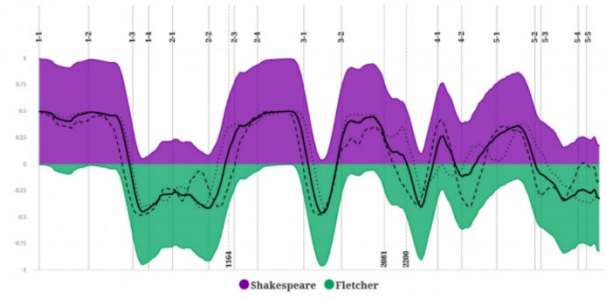

Figure 1MIT analysis of Henry VII

Purpose: The purpose of this project is to train a neural network to predict whether each scene in Edward III is written by Shakespeare or another author. The idea builds upon prior explorations of Shakespeare's works, as noted in the background portion of this paper. To achieve this purpose, word embedding and the recurrent neural network architecture were used to analyze and classify the text.

Data: The data used for this project is plays written in the late 16th and early 17th centuries, also known as the Elizabethan era of plays. The first portion of the data consists of Shakespeare's works. In particular, the plays selected are all of Shakespeare's histories, which is a style of play that deals with historical characters and comments on contemporary society. This category of play, often referred to as the *First Folio*, was used because it appeared more likely to use consistent language and style to tell a story compared to the wide array of all of Shakespeare's works. All these works follow iambic pentameter, a style of poetry where each line consists of ten syllables and is read in a set rhythm.

The other component of the dataset, which was harder to assemble, consists of works by other prominent playwrights and poets of the same time. This includes John Milton, Thomas Kyd, Ben Jonson, John Webster, and Christopher Marlowe, all of whom published works in the Elizabethan era and had a connection to Shakespeare and/or his works. Importantly, the plays selected for this portion also follow iambic pentameter because that is likely to affect the way the author writes the story and the intent for the data is to analyze a work written in that style.

Both components of the dataset were then preprocessed using the same technique. Each line from the plays was considered a singular datapoint. First, all lines with less than 30 characters were eliminated, as these lines were extremely likely to either be stage directions or just the names of whichever characters was about to speak. After this, all the digits were removed, and the letters were all converted to lowercase for consistency. Last, each line was assigned either a 1, for Shakespeare, or 0, for not Shakespeare, as a label. This final set of data was shuffled and input into a *.csv file. The test data, which is Edward III, was processed by each individual scene in the same manner and then saved into an individual *.csv without labels (as there is no definitive classification).

The resulting training dataset has a size of 44,578 lines, 29691 of which are Shakespeare and 14888 are not.

Within TensorFlow, the data is further processed for training/testing. Borrowed from the methods used to create word embeddings on the IMDB sentiment analysis dataset, the data is first tokenized; the 10000 most common English words found in the data are assigned an index number. The array of word indexes, each of which correspond to a line in the dataset, is then padded to ensure size uniformity. This final processing creates the data that is input into the model for training.

Measuring Success: Before the models are discussed, there needs to be an understanding of what is considered success. The first measure of success is the model beating the naïve baseline of the model, which is 66.6% because the Shakespeare data makes up 66.6% of the total training data. Thus, if the model is beating this baseline number when it is predicting on validation data, a level of confidence exists that the model can discern Shakespeare's works from others.

The second measure of success is vaguer. Because of the nature of the project, where the final model is applied to data without a known and definitive answer, it is hard to compute an accuracy based on how well it performs on Edward III. To address this, the output of the model's predictions of each scene will be compared with other, non-neural network predictions for the composition of Edward III. For the sake of brevity, the focus of this portion of analysis will be a writeup by BBC that sources scholars to classify Edward III by each scene. This measure should provide an interesting comparison between a technical and non-technical analysis of the same work.

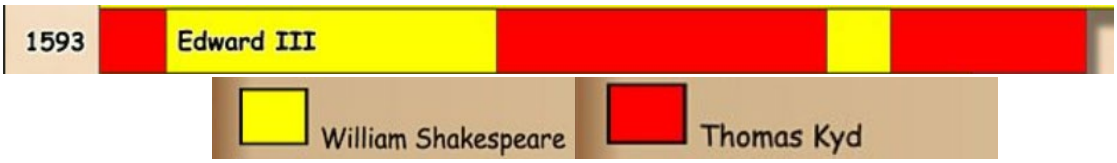

Figure 2 BBC analysis of Edward III

Baseline Model: The baseline model for this project consists of an embedding layer followed by 3 layers of TensorFlow's SimpleRNN, with 16, 32, and 16 neurons per respective layer. After these layers is two dense layers with 32 and 1 neuron, respectively. Upon running the baseline model, the immediate problem is overfitting. Within 3 epochs of the first run, validation accuracy hit 0.8548 while training accuracy climbed to over 0.9400. This shows that follow on models should focus on addressing mitigating overfitting while maintaining or beating this baseline accuracy.

```
Embedding(vocab_size, embedding_dim, input_length=max_length),
SimpleRNN(16, return_sequences=True),
SimpleRNN(32, return_sequences=True),
SimpleRNN(16),
Dense(32, activation='relu'),
Dense(1, activation='sigmoid')
```

Figure 4Architecture of the baseline model

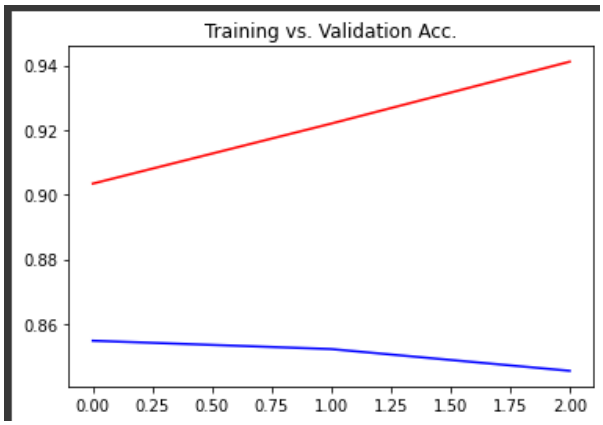

Figure 3 Training versus Validation Accuracy for the Baseline Model

Exploring More Complex Models: The next step in the project was to create more complex models that would combat overfitting (we already have a model that overfits). To do this, 3 types of RNN layers were explored. Model 1 builds upon the SimpleRNN baseline, but incorporates both recurrent dropout and dropout layers. This model achieves a slightly lower validation accuracy of 0.8492, but demonstrates less extreme overfitting and is able to train for more epochs before callbacks are triggered.

```
Embedding(vocab_size, embedding_dim, input_length=max_length),
SimpleRNN(16, return_sequences=True, recurrent_dropout=0.2),
Dropout(0.3),
SimpleRNN(32, return_sequences=True, recurrent_dropout=0.2),
Dropout(0.3),
SimpleRNN(16, recurrent_dropout=0.2),
Dense(32, activation='relu'),
Dense(1, activation='sigmoid')
```

Figure 4Architecture of the regularized SimpleRNN model

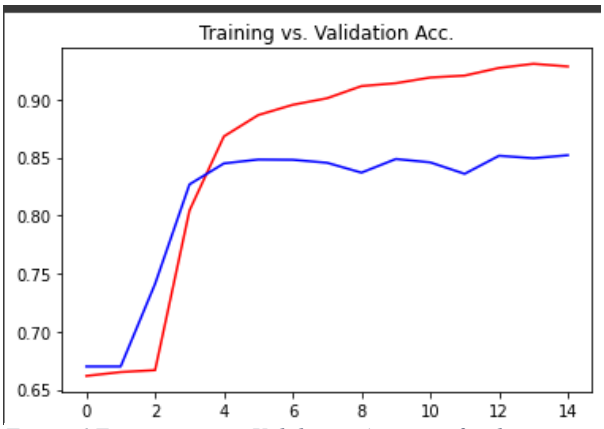

Figure 6 Training versus Validation Accuracy for the regularized SimpleRNN Model

Model 2 of the exploration employs TensorFlow's Gated Recurrent Unit, or GRU, layer. This model, without regularization, demonstrated a similar level of overfitting as the baseline. To combat this, the final model incorporates recurrent dropout, recurrent l1 and l2 regularization, and inter layer dropout. This model peaks at a validation accuracy of 0.8521.

```
Embedding(vocab_size, embedding_dim, input_length=max_length),
GRU(32, return_sequences=True, recurrent_dropout = 0.2),
Dropout(0.3),
GRU(32, return_sequences=True, recurrent_dropout = 0.2),
Dropout(0.3),
GRU(32),
Dense(32, activation='relu'),
Dense(1, activation='sigmoid')
```

Figure 6 Architecture of the GRU model

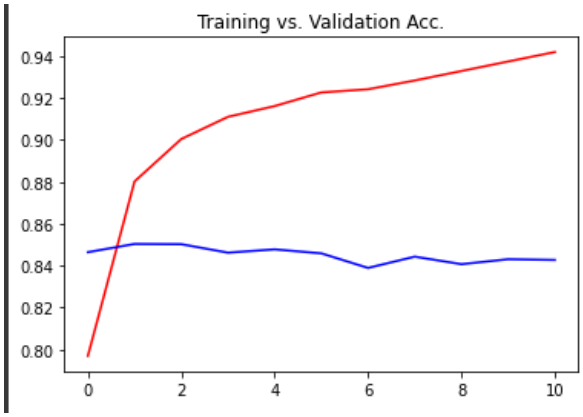

Figure 5 Training versus Validation Accuracy for the GRU Model

Model 3 of the exploration employs the Long-Short Term Memory, or LSTM, architecture. With this model, the simple baseline does not show the same validation accuracy that the previous two architectures do and struggles to beat the naïve accuracy. While overfitting has not become a problem with this model, the low validation accuracy across different numbers of layers and neurons means it is not the optimal approach for this problem.

```
Embedding(vocab_size, embedding_dim, input_length=max_length),
LSTM(32, return_sequences = True),
LSTM(32, return_sequences = True),
LSTM(32, return_sequences = True),
Dense(32, activation='relu'),
Dense(1, activation='sigmoid')
```

Figure 7Architecture of the LSTM model

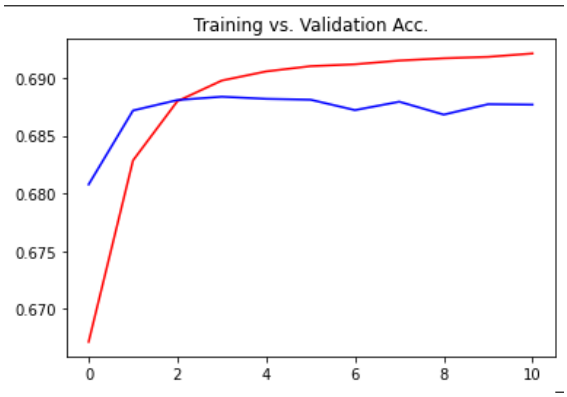

Figure 8Training versus Validation Accuracy for the LSTM Model

Results: Due to Model 3 barely beating the naïve baseline, the results will be analyzed for Model 1 and 2. Below is a plot of the confidence that each model has that a scene is written by Shakespeare. The higher the percentage assigned to each scene, the more confident the model is that it was written by Shakespeare. The red line, which is at 0.5, indicates the cutoff for if the model predicts the scene is written by Shakespeare. A value over 0.5 is a prediction that the passage was written by Shakespeare and a value below 0.5 is a prediction that the passage was not written by Shakespeare. Both models being employed (RNN and GRU) have similar predictions.

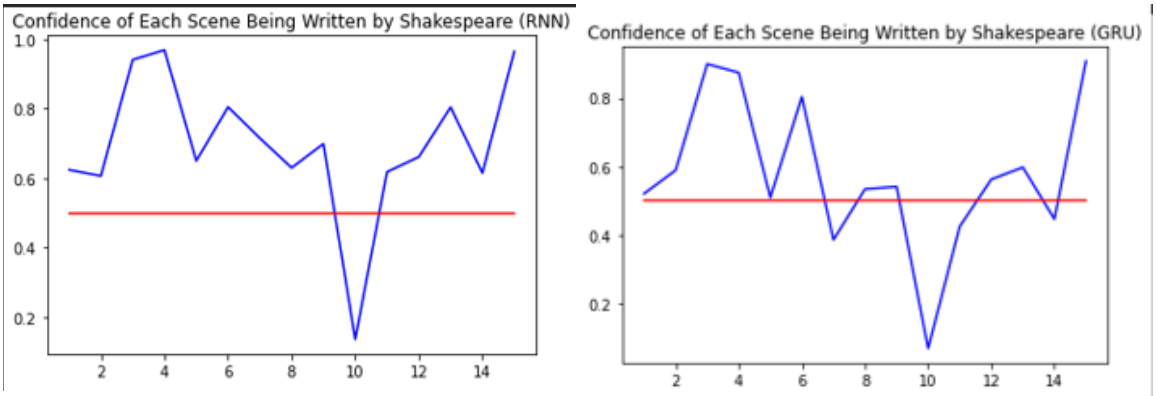

Figure 9Predictions from the SimpleRNN and GRU models

And, in line with the second measure of success mentioned earlier, these results can be compared with the BBC analysis of Edward III.

Figure 10BBC Analysis of Edward III

There are some interesting similarities and differences between the analyses. First, the large Thomas Kyd section in the middle of the BBC graph, around scene 10, follows the same prediction that the models have. The models around this scene demonstrate a relatively low level of confidence that Shakespeare is the author of these sections (with some scenes dipping below the 0.5 threshold). Additionally, the major chunk of Shakespeare found after the first scene in the BBC analysis corresponds to higher levels of confidence in the model predictions.

A point of difference between the BBC analysis and the model analysis is scene 15, which BBC classifies as being within a large chunk of Thomas Kyd writing, has one of the highest levels of confidence for each model's predictions.

Ultimately, the models and anecdotal analysis by BBC are very similar and validate, to a reasonable extent, models 1 and 2.

Takeaways: While there is currently not a way to get a complete test on the accuracy of the model when dealing with Edward III, the validation results and anecdotal analysis indicates that there is some accuracy to the model. Looking forward, if I had more time, I would like to build a more robust dataset with more non-Shakespeare writing. A limitation to this was finding texts that were digitized to build the dataset with. Additionally, I would like to experiment with pretrained models that deal with the English language and apply that to Edward III to compare my findings. During my presentation, the question was asked about generating my own non-Shakespeare data. This is a valid option, but that generation would be a separate project and difficult to classify what the end state would be; i.e. is the goal to generate text for the dataset that is virtually indistinguishable from Shakespeare or random noise that is "good enough" to not be Shakespeare.

Overall, I am happy with the way the models turned out and I learned a lot about using neural networks outside of classroom datasets and I learned a lot about the history of Shakespeare. Additionally, I will be sharing my findings with one of my mentors, who is a Milton and Shakespeare scholar who took interest in this project when I shared it with him.

Works Cited

Aguilar, K. (2021, December 15). *NLP Techniques with Shakespeare's Plays - Analytics Vidhya*. Medium. https://medium.com/analytics-vidhya/nlp-techniques-with-shakespeares-plays-d8843ba26a4f

arXiv, E. T. F. T. (2020, April 2). *Machine learning has revealed exactly how much of a Shakespeare play was written by someone else*. MIT Technology Review. https://www.technologyreview.com/2019/11/22/131857/machine-learning-has-revealed-exactly-how-much-of-a-shakespeare-play-was-written-by-someone/

Babillard, A. (2000, August 10). *Shakespeare's Missing Plays: Edward III*. Shakespeare Online. http://shakespeare-online.com/plays/edwardIII.html

BBC. (2016, August 18). *Shakespeare Lives - Who Wrote Shakespeare?* https://www.bbc.co.uk/programmes/articles/211LBPTmBYp2rbh4bSQlSTS/who-wrote-shakespeare

Cotra, A. K. (2021, December 14). *Sentiment Analysis using Word embeddings with Tensorflow*. Medium. https://medium.com/swlh/sentiment-analysis-using-word-embeddings-with-tensorflow-ab8eab7641a4

Jamieson, L. (2019, September 11). *How to Identify a Shakespeare History Play*. ThoughtCo. https://www.thoughtco.com/shakespeare-histories-plays-2985246

Wikipedia contributors. (2022, March 28). *Shakespearean history*. Wikipedia. https://en.wikipedia.org/wiki/Shakespearean_history