SI 670 Fall 2020 Project: "Hear Ye Hear Ye: Machine Translation and Sentiment Analysis of Modern Literature into Early English Language"
Team: Maggie Brodbeck, Marilu Duque

Introduction

The English language has evolved over the ages as civilizations advanced and societies expanded. Language speakers required new ways to communicate their new ideas, and language adapted to fit that need. As language has changed, many of the sentiments and subjectivity expressed have shifted meaning as well. To better understand this evolution, our project aims to translate existing modern literature into early English using machine learning (ML) methods. Additionally, we aim to analyze the sentiment of the modern literature before and after translation into early English. This will help us better understand how sentiment perception has evolved as the English language has evolved.

This problem is important because it gives us a clearer understanding of sentiment analysis throughout history and helps us acknowledge public perceptions of certain topics throughout time and how they have evolved. This knowledge about change in perception is useful in planning and strategizing for future events. This understanding is important as it would help academic scholars better assess key context clues of literature during those times. Additionally, this could help researchers assess how certain literature would have been perceived during specific eras in history. Currently, ML projects within early English translations, heavily rely on user input rather than translation conducted through existing pieces of literature. Additionally, there are few research projects focused on the nexus of converting modern literature into early English language and then analyzing its sentiment.

To tackle this problem, we work to train our models on an early English dataset consisting of Shakespeare's Romeo & Juliet. To test our translation, we are utilizing the modern literature dataset Harry Potter and the Sorcerer's Stone from the critically acclaimed Harry Potter novel series. Using both these world-known pieces of literature will help us better understand how sentiment and subjectivity have evolved over time and through the evolution of popular literature of its time.

Methods

Algorithms Used:

The machine translation model utilized deep learning algorithms. It uses a Bidirectional Long Short-Term Memory (LTSM) algorithm. This recurrent neural network (RNN) was chosen as it is typically used for machine translation models due to its effectiveness in learning sequence-to-sequence order, especially with the Bidirectional LSTM since it processes data in both directions.

The sentiment analysis models utilized a preliminary Bag of Word (BoW) model that was then applied to Logistic Regression (LR) and Naive Bayes (NB) models. The BoW model was chosen due to its specialized ability to process textual data for machine learning. The LR model was chosen due to its unique ability to both classify and predict outcomes of assigned data. Additionally, LR pairs well with the outcomes of a BoW model. The NB model was chosen due to its focus on independence between features and its usually higher accuracy results.

Before any models can be applied, the modern literature undergoes a text preprocessing which includes splitting text by individual word, lowering cases, removing punctuation and stop words. Particularly, the stop words must be removed in order to reduce noise in the data. The BoW was applied using the CountVectorizer tool which transforms the text into a collection of tokenized vocabulary. This is used to encode the text and prepare it for splitting of testing data for the aforementioned models. Once this was done, the training and test data were split and applied to the LR and NB models. Additionally, extra analysis was conducted by applying the TextBlob natural language processing (NLP) python library to better understand sentiment.

The machine translation model employs the original and modern version of *Romeo & Juliet.* The text data files are opened separately and assigned their own variables. Both datasets undergo several steps of preprocessing. At the time the datasets are read, the lower() and split() are applied to each line. A function defined as re_clean() loops through each line in the data to add spaces to pad punctuation and strip trailing and leading characters.

Code Description:

After applying re_clean() on both texts, a function defined as token() is created. In this function, a Keras Tokenizer() is created and fit on the texts passed into it. The text is then converted to sequences using text_to_sequences(). The sequences are then padded at the end using pad_sequences() and padding='post'. When applied to the data, the function returns both the tensor and tokenizer. The maxlen='' parameter is applied to the needed text as well to ensure the tensors are the same length.

The input tensor and the output tensor are then passed to sklearn's train_test_split function to create training and testing data. The test_size is set to 0.2. The function produces four sets: input_tensor_train, input_tensor_test, target_tensor_train, target_tensor_test.

After the preprocessing is completed and the training and testing sets are created, an empty Sequential() model is initialized. An Embedding() layer is added first. The input dimension is the length of the input token's word_index, and the output dimension is the length of the out token's word_index. +1 is added for padding. The input_shape corresponds to the target tensor.

The next layer is a Bidirectional(LTSM()). An arbitrary number of units is passed. Return_sequences is set to True, and the activation is set to 'tanh'. Next, a dense layer is added with 64 units, which corresponds to the batch size when the model is later fitted. The activation is set to 'relu'. The last layer is another dense layer. The number of units is the length of the word index+1, and the activation is again set to 'relu'.

The model is compiled with 'sparse categorical cross-entropy' loss and 'adam' optimizer. The metric is set to accuracy so that the model can be evaluated. The model is fit to training data with the testing data as the validation data. The batch_size is 64, and the epoch is 10. When the model is finished fitting, a function id_to_text is defined. The target token's word_index is first converted back to words. Then the range of the logits is looped through, and the input text is converted with the predictions. After the function is defined, the model predictions are then passed into the function to produce translation.

The sentiment analysis models are run for efficiency with the Harry Potter and the Sorcerer's Stone .txt. File. Textual preprocessing is performed in order to ensure the best model accuracy. This processing includes tokenization using the Natural Language Toolkits (NLTK) word_tokenize function. Then all words are converted into lowercase, .lower(), punctuation is removed using a for loop only checking for values accepted by .isalpha(). Lastly, English classified stop words are removed using a for loop and the NLTK corpus. As described previously, we then apply our pre-processed text into our BoW model and then the LR and NB models.

Data:

While the initial proposal consisted of two datasets with two texts each, the data had to be reduced in order to process the experiments on our machines and mitigate potential resource starvation. In addressing this problem, we were aware that it would likely affect the efficiency and accuracy of our models.

The sentiment analysis model was trained using individually tokenized words from the Harry Potter text which was processed through the BoW model. The model utilized the text to create its own trained vocabulary. The Harry Potter and the Sorcerer's Stone dataset was acquired from a GitHub repository created by Formcept, a data analysis company in India. The text consists of the entire book's corpus totaling 459,169 words in the dataset.

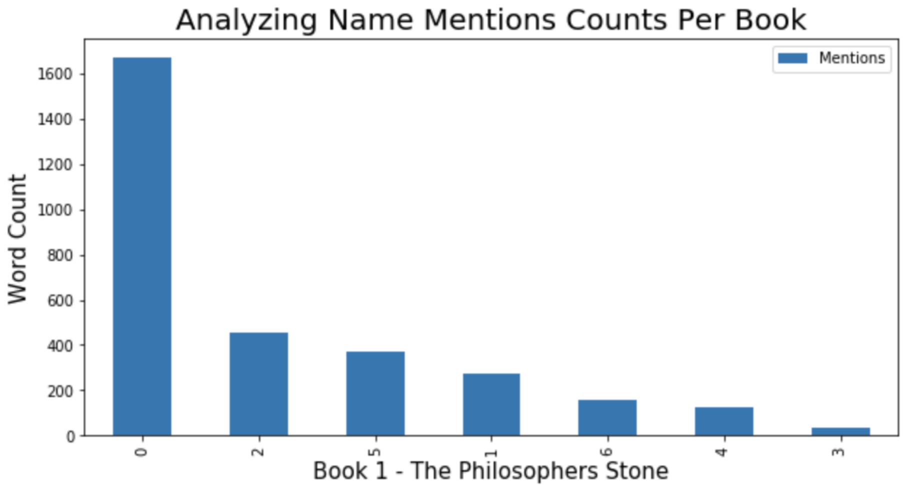
The machine translation model utilized two pre-existing datasets from the paper "Paraphrasing for Style" (Xu, et al). It can be found on Github. The first dataset is a 943 line sampling of the original *Romeo & Juliet* in early modern English**.** The second dataset contains the corresponding modern-day translations.

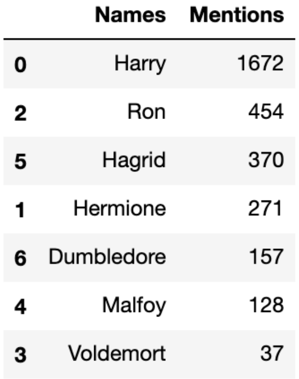
Evaluation and Analysis

For our project, we employed an experimental research design methodology to explore the texts and best ascertain the models that would best suit our datasets.

In order to best understand the sentiment of the original Harry Potter text with hopes of comparing it to a translated final copy, it was important to perform an exploratory analysis. This analysis showed the most popular characters/names, most popular houses, total vs. unique word count, and a word cloud of popular words.

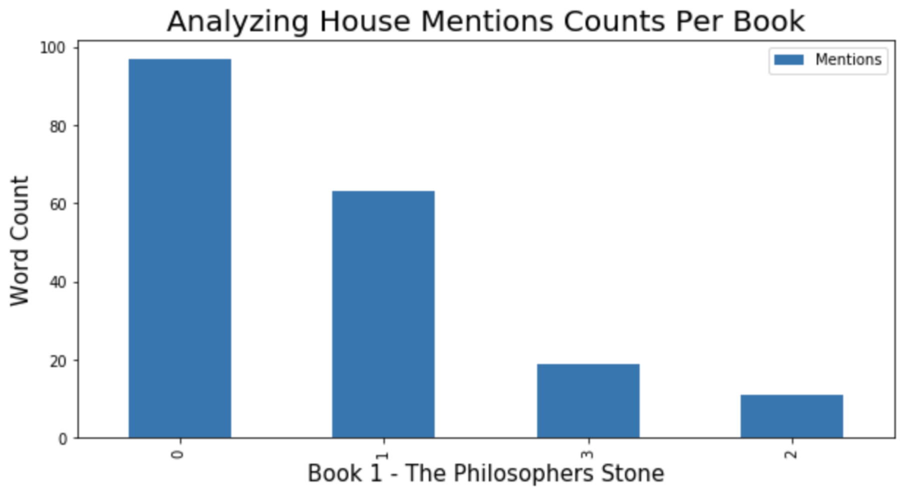
Most Popular Characters:

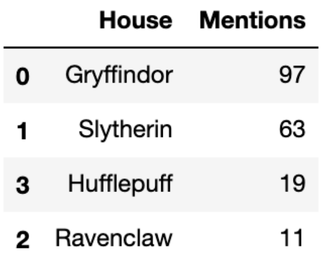




	Names	Mentions
0	Harry	1672
2	Ron	454
5	Hagrid	370
1	Hermione	271
6	Dumbledore	157
4	Malfoy	128
3	Voldemort	37

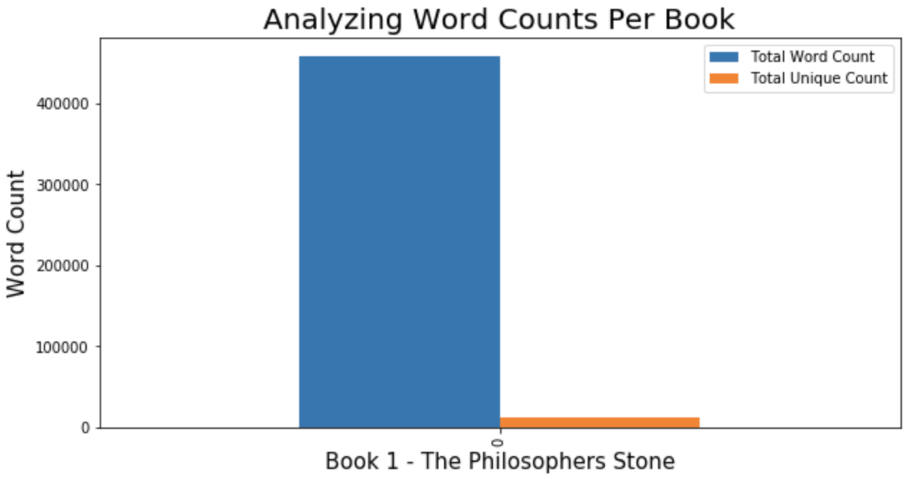
Most Popular Houses:

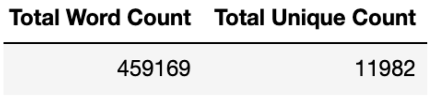




	House	Mentions
0	Gryffindor	97
1	Slytherin	63
3	Hufflepuff	19
2	Ravenclaw	11

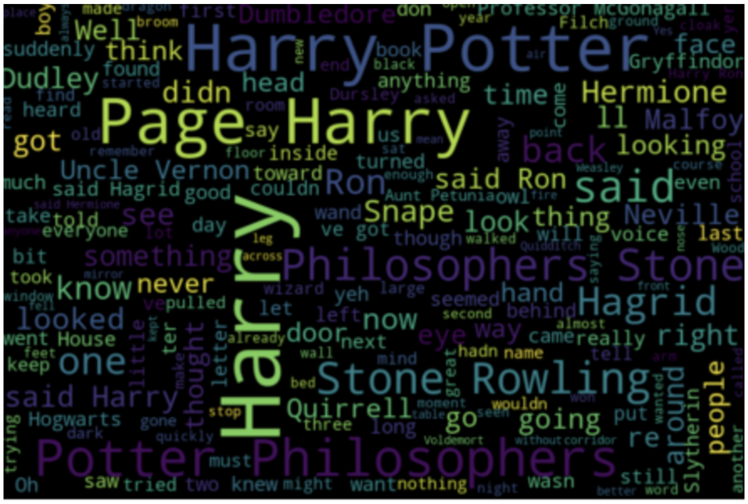
Total Vs Unique Word Count:





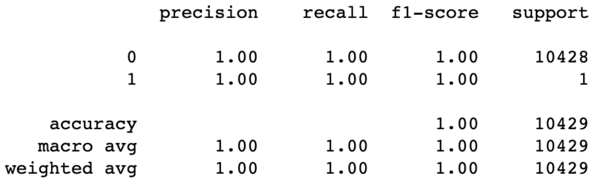
Total Word Count	Total Unique Count
459169	11982

Word Cloud:



Results:

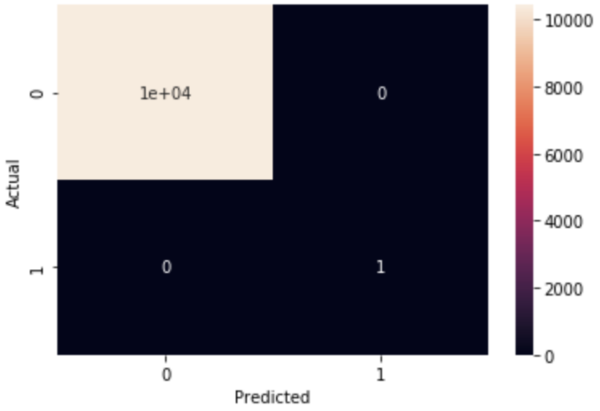
Once we did an exploratory analysis of the modern literature, we pre-processed the text and applied the models. These models assessed the data with a testing shape of (10429, 5477) and training shape of (31285, 5477). With this, our results noted that after dealing with potential errors in both the dataset and overall coding, the LR and NB accuracy scores matched even though the approaches were different.



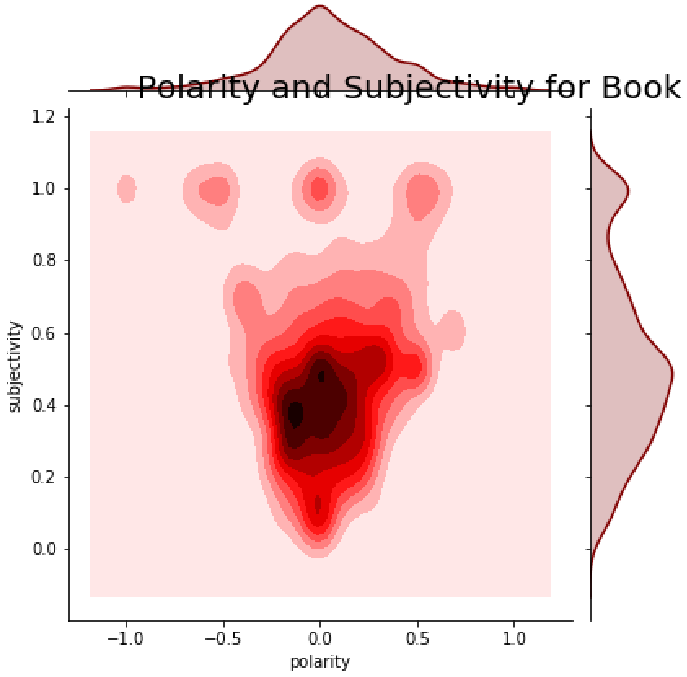
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     10428
           1       1.00      1.00      1.00         1

    accuracy                           1.00     10429
   macro avg       1.00      1.00      1.00     10429
weighted avg       1.00      1.00      1.00     10429
```



To gain a better understanding of our text and advance our sentiment analysis model of the modern literature text, we employed the use of TextBlob. This tool allowed us to ascertain the sentiment and subjectivity expressed through the literature.



Through the plot above, we can see that the average subjectivity expressed in the text is .475, which is slightly above neutral subjectivity due to its close nature to +1. The polarity or overall emotions in the text resulted in a much more neutral score at .050. The subjectivity was expected due to the emotional topic nature and audience of this text, but we hypothesized the polarity to be higher for the same reason. In looking at the joint plot, we can see distributions and peaks in the graphs. Specifically, polarity seems more stable and neutral while subjectivity can be perceived as fluid.

Final Model: Sequential

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 57, 2307)	5322249
bidirectional_1 (Bidirectional)	(None, 57, 512)	5251072
dense (Dense)	(None, 57, 32)	16416
dense_1 (Dense)	(None, 57, 1854)	61182

Total params: 10,650,919
Trainable params: 10,650,919
Non-trainable params: 0

Within the machine translation model, in addition to experimenting with multiple model types and algorithms, the machine translation model was evaluated and analyzed at several steps throughout the final design. The goal of the model was to ultimately produce a translation of the original version to a modern version and then use that model to convert *Harry Potter* into Shakespearean English. In addition to analyzing the translation, the accuracy scores, loss, and BLEU scores were also to be evaluated.

Samples from model prediction training:

Input text	Target text	Predicted text
as mine on hers , so hers is set on mine , and all combined , save what thou must combine by holy marriage .	we re bound to each other in every possible way , except we need you to marry us	you me the <end> you , , , , <start> you you you sucked
then plainly know my heart s dear love is set on the fair daughter of rich capulet .	i love rich capulet s daughter .	will to i . to to i . the i , i . .
riddling confession finds but riddling shrift .	a jumbled confession can only receive a jumbled absolution .	<start> <start> <start> . don don don

Despite producing higher accuracy scores (~80%), low loss (~1.00), and undergoing several rounds of hyper tuning, the experimental machine translation model could not produce a coherent translation and struggled to map the input sequences to the target sequences.

Samples from model prediction testing:

Input Text	Predicted Text
were proud to say that they were perfectly normal,	… ….
The Dursleys had everything they wanted, but they also had a secret,	You , , you , you <start>

Related Work

Currently, machine learning projects within early English translations heavily rely on user input rather than translation conducted through existing pieces of literature. Additionally, there are few research projects focused on the nexus of converting modern literature into early English language and then analyzing its sentiment. Some existing work in this field includes “Sequence to Sequence Learning with Neural Networks” (Sutskever, et al) which addressed the use of machine translation for translating English to French, and “Paraphrasing for Style” (Xu, et al) which addressed using machine translation for paraphrasing Shakespearean works. Both works are discussed below.

Several publications concerning sequence-to-sequence machine translation exist, though nearly all are concerned with translating one language to another, like Spanish to English. The article “Paraphrasing for Style” tackles the previously unaddressed problem of converting Shakespeare to modern English with machine translation and was the first and major publication to explore this type of work. The research team focused on ‘paraphrasing language while targeting a particular writing style (Xu et al.).’ The project was successful in training its model and were able to produce exact translations. They trained the model using the original and modern version sampling of several Shakespearian works, which is the project from which we pulled our data. Nearly all tutorials concerning Shakespearian machine translation use this data as well.

The previously mentioned model differs in intent from our model that is focused on translating an earlier version of English to modern English. A major paper that is focused on language translation is “Sequence to Sequence Learning with Neural Networks” (Sutskever, et al). This paper first addressed the use of deep learning for sequence to sequence, which was not used for machine translation before. The model used LTSM and was successful in translating their data, regardless of sentence length. The team also found that reversing the order of the input sentences improved the model’s performance as well due to the introduction of short-term dependence between the data.

Discussion and Conclusion

Overall, the machine translation model produced high accuracy scores during training, but it performed poorly in producing direct translation results. The model had difficulty in matching target sequences beyond the first few parts of the input index, which may be improved with more data, layers, and epochs. With stronger computer processing and data, it would likely produce a more accurate performance. The sentiment analysis models produced high, but identical accuracies which begs the question of potential embedded errors. More time to work on these would help ascertain these errors. Additionally, the TextBlob tool showed the modern literature piece (before any translation) to be near neutral polarity and above neutral subjectivity which is to be expected in the literature that tackles the topics of life, death and nature can often stir up a vast array of emotions throughout its wording and characters.

Throughout this project, we learned about tuning models, preparing textual data for model applications, and how to scope projects so they are more specific and less resource-intensive on our devices. This project showed us how to balance the various moving parts in a machine learning project and communicate challenges effectively with our team. The research conducted in designing this project also showcased the lack of research being done in the early and modern literature NLP space. We hope our projects adds to the limited existing project base and inspire future technologists to create further. Additionally, we were able to apply both new and known machine learning algorithms and best practices for creating predictive models. This project helped expand our machine learning knowledge, adding to a team member's previous NLP project, and incorporating our love for literature into something tangible from a data science perspective.

If we had more time to continue working on this project, we would want to try different models to ensure the best accuracy and completion of the training and translation of the Shakespearean text. Given more time and resources, added training of early English literature through more textual datasets would help ensure a better result.

References

Papers:

Brownlee, J. (2017 December 20). *A Gentle Introduction to Calculating the BLEU Score for Text in Python.* Machine Learning Mastery. machinelearningmastery.com/calculate-bleu-score-for-text-python

Chollet, F. (2017 September 29). *Character-level recurrent sequence-to-sequence model.* Keras Documentation. keras.io/examples/nlp/lstm_seq2seq/

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. (2014). *Sequence to sequence learning with neural networks*. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14). MIT Press, Cambridge, MA, USA, 3104–3112.

TensorFlow. (2020 September 26). *Neural machine translation with attention.* TensorFlow Documentation. tensorflow.org/tutorials/text/nmt_with_attention

Xu W., Ritter A., Dolan W.B., Grishman R., Cherry C. (2012). *Paraphrasing for style*. 24th International Conference on Computational Linguistics - Proceedings of COLING 2012: Technical Papers, pp. 2899-2914.

Code References:

TensorFlow/Keras Documentation - https://www.tensorflow.org/api_docs/python/tf/keras
Machine Translation - https://github.com/santanu94/Machine-Translation-Udacity-NLP
TextBlob - https://textblob.readthedocs.io/en/dev/
NLTK - https://www.nltk.org/

Data:

Rome & Juliet - https://github.com/cocoxu/Shakespeare/tree/master/RomeoJuliet
Harry Potter -
https://github.com/formcept/whiteboard/tree/master/nbviewer/notebooks/data/harrypotter