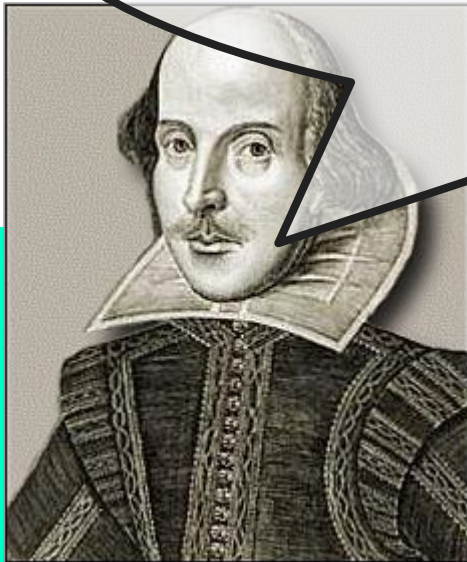


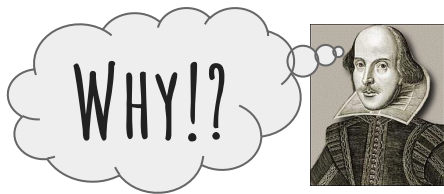
ARE YOU SHAKESPEAREAN?



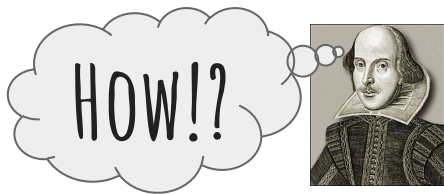
**Generating and Classifying Shakespearean
Text with Neural Networks**



- Generate text that sounds Shakespearean from a data model sourced from Shakespeare's complete works
- Classify text as "Shakespearean" (positive class) or "Amateur" (negative class) based on a data model trained on Shakespeare's complete works and works of other authors
- Evaluate generated-text model based on classification model. How well does the generating model generate Shakespearean text?

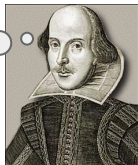


- Short answer: Shakespeare is awesome
- Underlying (and overly ambitious) goal was to create a poetry generator that could recite poetic verse.
 - Even more underlying (and impossibly ambitious) goal was to create a language generator that could imitate actual language but be a different language.
- Turns out poetry and data science are not strangers – research in applications of neural networks to poetry generation has been ongoing since 2018 (or earlier)
- So... limited to generating text that sounds Shakespearean



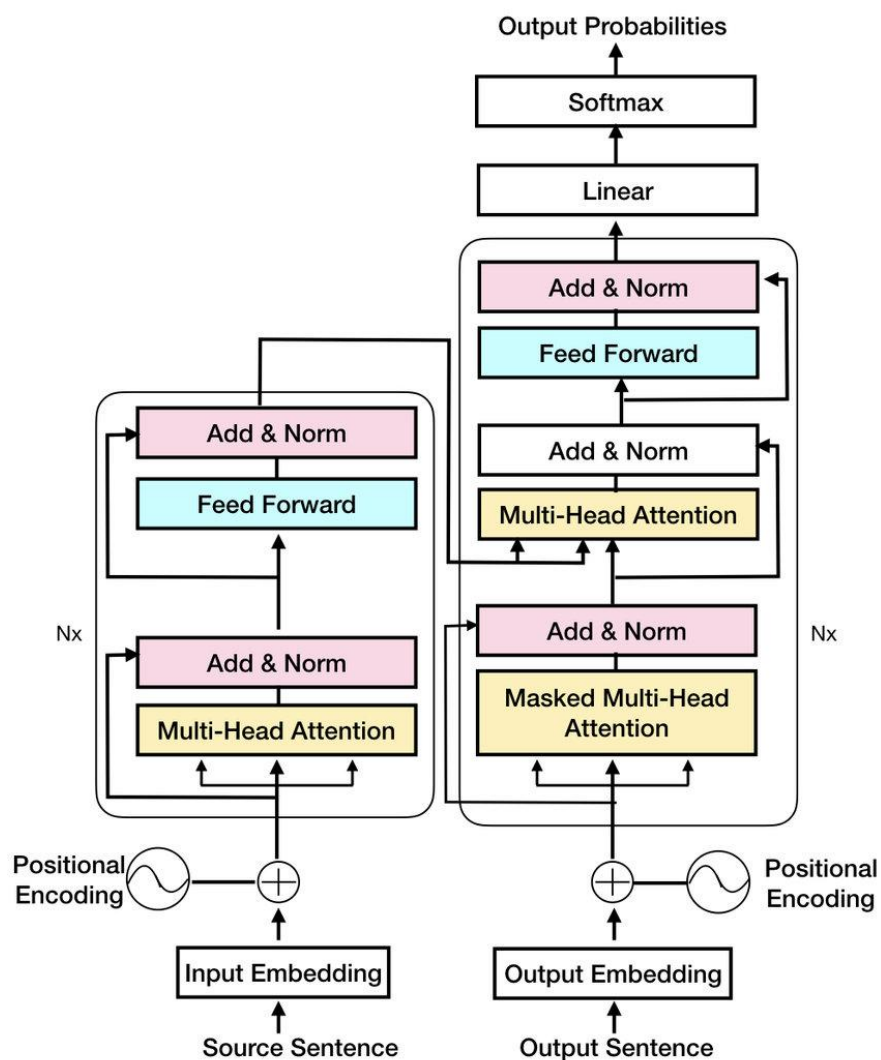
- Answer: Transformer Models
 - Models based on neural networks with interacting layers
- Many flavors for language modeling:
 - Causal Language Modeling: like time series modeling for arbitrary sequences - predict next item in sequence based on previous items
 - Masked Language Modeling: considers context of sequence to fill in blanks in sequence
 - Can also be used to classify sequence based on context
- My review of research suggests that much of the success of these models comes from proper data cleaning and categorization before feeding the models

TRANSFORMERS!?

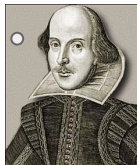


- Neural-network-based models using encoder-decoder architecture with interacting layers depending on use-case.
- Causal LM uses decoder only.
- Classification and others use encoder and decoder portions with “attention” to identify context

Image from: Hasan, Md. Arif & Alam, Firoj & Chowdhury, Shammur & Khan, Naira. (2019). Neural Machine Translation for the Bangla-English Language Pair.

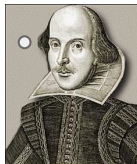


IMPLEMENTATION!?



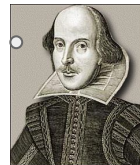
- 2 models: (1) causal and (2) classification.
- Transfer learning: pre-trained models were fine-tuned to specific text
- Causal model: decoder-only, predicts next word(s) based on previous words
 - Used GPT2 (distilgpt2) pre-trained model fine-tuned on all of Shakespeare's works, split by sentence.
- Classification model: encoder-decoder model - takes context into account to classify a sequence of text data
 - Used BERT (distilbert) pre-trained model fine-tuned on Shakespeare's works as well as those of other period-relevant and -not-relevant authors.

CAUSAL MODELING!?



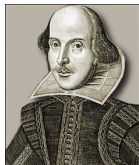
- This is the fun stuff
- Generate new text from prompts
 - Transfer learning - used pre-trained model as back-end, fine-tuned on specific data (Shakespeare)
 - Fine-tuning pre-trained model probably serves a good purpose in bridging gap where prompt might not match any text in source
- Data: all of Shakespeare's works, split into sentences
 - Total count: 76,578 sentences

CLASSIFICATION MODELING!?



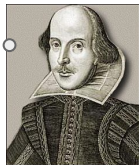
- Classify text based on learned data
 - How can I determine if I'm generating Shakespearean text?
 - Obviously, teach the model to discern Shakespearean text from other text!
- Data:
 - Total: 113,000 sentences of text (almost 8M chars)
 - Feature balances:
 - 76,578 sentences Shakespeare
 - 36,809 sentences other authors including approx 4000 sentences of GPT2 generated text
 - Ratio: 68% Shakespearean, 32% other
 - Baseline accuracy was therefore 68%

RESULTS!?



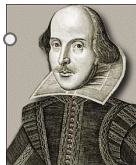
- Causal Model: text reads generally in a somewhat-Shakespearean style
 - Perplexity metric can be used here - was not used due to library dependency issues
 - Played it by ear to start, but later...
- Classification Model: usual metrics apply
 - Data split into train and test sets
 - Train: 100,917 sentences
 - Test: 11,337 sentences
 - Model trained on train set, metric determined from test set
 - Usual classification metrics: accuracy, confusion matrix, etc.
 - Achieved 92% accuracy on the model!

CAUSAL EVALUATION!?



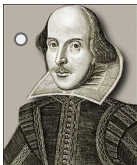
- Ran classification model on 3 sets of generated text:
 - Wine descriptions hosted on Kaggle
 - Shakespeare's complete works (classification model positive class)
 - Other works (classification model negative class)
- Random samples from each set were selected, and a prompt was created to feed to the causal model
- Results:
 - Wines: 68% Shakespearean
 - Shakespeare: 96% Shakespearean
 - Other: 49% Shakespearean
- Shakespeare in, Shakespeare out – otherwise not bad!

DEMO TIME!?



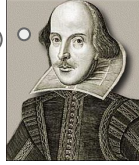
- Sure!

LIMITATIONS!?



- Classification accuracy dependent on text length.
 - Proper prompts require more text - single words, for example, can be used in too many contexts.
 - Metrics could use more analysis - what inputs tend to produce more Shakespearean outputs?
- Next steps:
 - Figure out how to classify rhyme. Probably can be rule-based.
 - Figure out how to classify meter. This is not trivial!
 - Generate multiple sentences composed of lines satisfying meter and rhyme requirements. Done! Poetry generator!

QUESTIONS!?



- Questions?



- GPT2 model: <https://huggingface.co/gpt2>
 - See also distilgpt2 model: <https://huggingface.co/distilgpt2>
- BERT model: <https://huggingface.co/bert-base-uncased>
 - See also distilbert-base-uncased model:
<https://huggingface.co/distilbert-base-uncased>
- TensorFlow Transformers tutorial:
<https://www.tensorflow.org/text/tutorials/transformer>
- Many relevant academic papers from [Google Scholar](#)