Machine Learning & Data Mining
Miniproject 2

Caltech CS/CNS/EE 155
Released February 17^{th}, 2017

---

1 Introduction

- **Group members:** Enrico Borba, Claire Goeckner-Wald
- **Team name:** Papa Mart's Mini Gary - The Comeback
- **Division of labour:** Enrico Borba: Programming, ideas, report visualization. Claire Goeckner-Wald: Programming, ideas, report assembly.

2 Pre-processing

- **Handling the dataset**
  - **Quatrains versus couplets:** Initially, we thought that we should train two different models, one on lines from the quatrains, and one on lines from the couplets. This way, we could more accurately capture the "shift in tone" that Shakespeare often performs during his sonnets. However, since we decided to 'force' rhyming using a rhyming dictionary garnered from the dataset, we ended up combining quatrains and couplets into one model.
  - **Punctuation:** we stripped the following punctuation from the sonnet data
- **The dictionary**
  - **Reason for use:** We used a dictionary to assign each word a unique number. We had to set all word to lowercase first, to avoid using "The" in the middle of a sentence, when we would rather have "the".
  - **Unexpected trouble:** The use of the dictionary is also one reason why we decided to treat lines from quatrains and couplets equally. Initially, we had one large dictionary that covered all words Shakespeare used in the dataset. As expected, some words were used only in the couplets, or only in the quatrains. However, we ran into errors when training two separate Hidden Markov Models, most likely because the model expected that if we gave it words (represented by numbers) $\{3, 15, 23, 14, 194\}$, that there would be $(1 - 194)$ states available. However, this was not the case.

3 Unsupervised Learning

- **HMM**
  - **Naive poem generation:**
  - **Package:**
  - **Number of hidden states:**

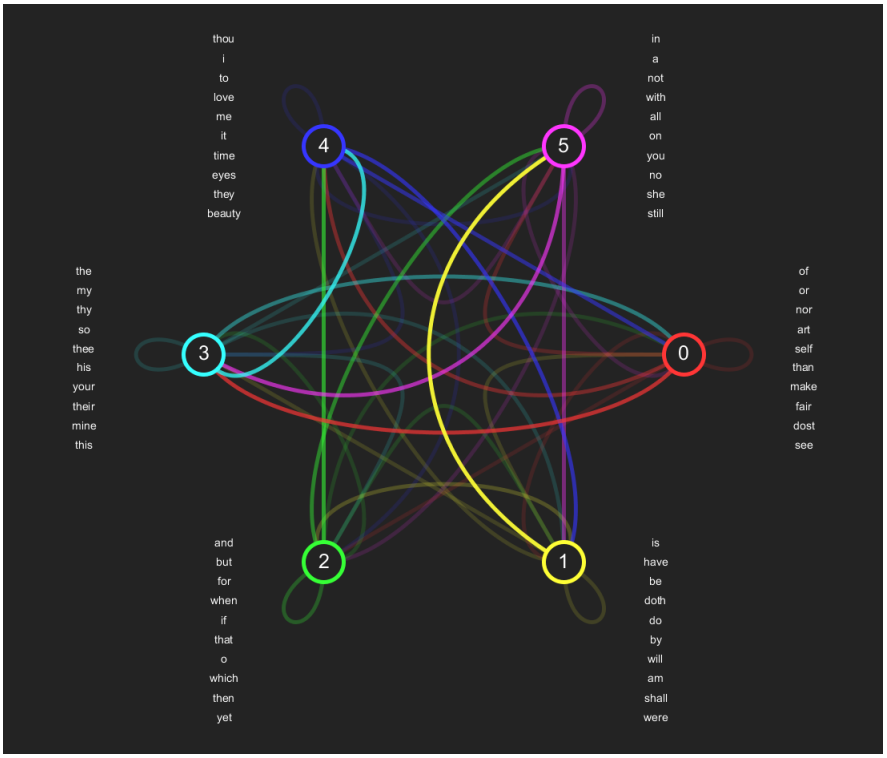
4 Visualization & Interpretation

- **Interpretations**

  - **Analyzing the model:** We tokenized the sonnets by words, removing punctuation. Because of this, the HMM can only determine patterns between words. Thus, the 6 HMM states have some complex pattern between the words, unlikely to be too related to the English grammar.

  - **Imagery:** The below image shows the 6 states and their transitions. Each state is colored to show directed transitions. That is, since state 1 is yellow, all yellow transitions come from state 1. More transparent transitions show lesser probabilities. So, the nearly opaque transition from state 1 to state 5 is of comparitively high probability. Also, the transitions are drawn in order of their probabilities. So, if some transition a appears above transition b, the transition a has higher probability than transition b.

    Furthermore, we have the top 10 words for each state represented next to the states. However, we disallow repititions of words in the image. If a word appears next to a state, then that state was the state that was most likely to emmit it.



- **State Analysis:** We notice that state 1 has a high collection of verbs. This continues as well: checking the top 20 words for state 1 yields "are, may, was, should, say, did, must, know, might, away", which is further composed mostly of verbs.

  State 1 has the highest probability of transitioning to state 5, which contains mainly prepositions and some nouns (which correctly follow from verbs).

  State 5 has the highest probability of transitioning to state 3, which has several possesive nouns. Inspecting the top 20 words of state 3 yields more possesive nouns such as "her, thine, our", and also some adjectives.

  State 3 then has the highest probability of transitioning to state 4, which contains even more nouns, as the sentence at this point, if began at state 3 (a likely verb), should reach the noun the verb is modifying.

  State 4 then has similar probabilities of transitioning to states 1 and 0, which allow for more compound phrases. Since the HMM mainly trained on single lines of the sonnets, taking more than 9 transitions almost guarantees a grammerless phrase. The average number of words per line in all of the sonnets is just over 8, thus the HMM has difficulty outputing coherent long phrases.

5 Poetry Generation

- **Algorithm**
  - **Example sonnet:**
  - **Quality of poem:**

6 Additional Goals

- **Topic**
  - **Subtopic:**
  - **Subtopic:**

7 Extra Credit: Recurrent Neural Networks

- **Topic**
  - **Subtopic:**
  - **Subtopic:**