Machine Learning & Data Mining Caltech CS/CNS/EE 155
Miniproject 2 Released February 17^{th}, 2017

SONNET SPOOFER
by Shari Kuroyama and Carly Robison

1 Tokenizing

Methods Used

What methods did you use and try to tokenize the sonnets?

- First we tried the naïve tokenization of using each word as a token. With enough states and iterations, this actually learned fairly well.
- We trained on individual lines from the poems. When we needed to generate the lines backwards from the rhymes at the end, we trained on individual lines read in backwards.
- Words were extracted by splitting on the spaces between them, so hyphenated words were included and every different spelling/conjugation of a word was counted separately.
- We kept apostrophes in words, but we removed periods, commas, colons, and parentheses.

Modifications

Did you have to make changes to the way you tokenized after running the algorithm and seeing the results? **Short answer: No.**

- We thought about splitting each word into syllables in order to keep to the 10-syllable format of a sonnet. However, we got around that by counting the number of syllables generated per line, and generating new words if the last word was too long.
- We also thought about trying to train using bigrams of words as tokens; but again, the naïve implementation worked fairly well.
- The naïve implementation was okay at getting meter, so we didn't see the need to supervise training to improve the meter.

2 Algorithm

Packages Used

We used the HW5 solutions for unsupervised HMM training. We also used `NLTK`'s dictionary of words to get syllable counts and the `pronouncing` package, which is based on NLTK's CMUdict, to find rhymes.

Parameters Used

- **Hidden States.** We experimented with different numbers of hidden states. For testing our other algorithms we used 5 hidden states, which produced poems with no sense at all. We found that the poems generated with 10 or 20 states were about the same level of coherence.
- **Number of Poems.** With fewer poems, we got more content coherence, but more training data the poems we generated had more grammatical sense.
- **Number of Iterations.** We found it hard to tell how many iterations to use; we used anywhere from 15 to 100 for our final poems, and this choice was almost exclusively based on runtime.

3 Poetry Generation

Generation Process

We iterated through multiple ideas to get a good sonnet. We first took 14 emissions of 10 lines each. We added rhyming by pregenerating pairs of rhyming words that both occurred in Shakespeare's sonnets, and adding them to the end of the lines in the ABABCDCDEFEFGG rhyming scheme. We eventually succeeded in "seeding" the HMM emission process with a hidden state chosen according to the emission probabilities for a given word. Our final poetry generation process trained on backwards lines and generated lines backwards from the rhyme at the end. We reversed the line and combined them to make a 14-line sonnet.

How to get it to look like a sonnet?

We used `pronouncing`'s functionality to ensure the correct rhyme scheme. We ensured 10 syllables by predicting sequences of new words when we ran out of space, and doing so until we could finish the line. We generated 14 lines.

Do they make sense?

We did not train or enforce meter, yet many of the sonnets have some kind of iambic meter, but not always carried through all of the lines. It is hard to tell from the most popular words for each state, but some states must have started contained stressed words and others would not have, and alternating between states would give the correct iambic stress pattern. The rhyme and syllable count are correct, because our generation process enforces their correctness. Most of the sonnets make sense if read dramatically, and because of the vocbulary still sound like Shakespeare. Here is the poem we submitted, which was actually generated with 40 states:

0

Seasons him make but heart my such being,
Rosy gardens all that although is morn,
Serving against self beauty i seeing,
Their name very so for and proof thy mourn,
Of by walls till that touches to the fixed,
In lawful who perceived cures whom and page,
All to all weed absent better betwixt,
Thy please should mayst hold outbraves to age,
And you which my us truly in art you,
Shall whose shapes but his and with my straying,
Which so dearest temptation i woe through,
What to that it cross not in be saying,
Black enough make muse thee like of thee halt,
Of not eyes flies false the viewest did fault.

Since all of Shakespeare's sonnets are numbered, we numbered our sonnet 0, because computer scientists zero-index.

4 Visualization and Interpretation

Hidden State Meaning

For at least 5 hidden states give a list of the top 10 words that associate with this hidden state and state any common features these groups.

Visualized transition Matrix, 10 states, 15 iters

```
state 0    words: ['the', 'but', 'when', 'that', 'my', 'as', 'then', 'which', 'if', 'for']
state 1    words: ['and', 'when', 'for', 'o', 'but', 'in', 'if', 'than', 'to', 'which']
state 2    words: ['the', 'of', 'a', 'my', 'to', 'that', 'on', 'this', 'as', 'thy']
state 3    words: ['that', 'the', 'my', 'to', 'in', 'what', 'are', 'a', 'not', 'as']
state 4    words: ['my', 'that', 'of', 'a', 'me', 'thee', 'do', 'the', 'no', 'have']
state 5    words: ['in', 'so', 'or', 'all', 'not', 'mine', 'like', 'do', 'those', 'will']
state 6    words: ['i', 'thou', 'thy', 'of', 'his', 'be', 'your', 'their', 'thee', 'with']
state 7    words: ['to', 'is', 'with', 'in', 'and', 'doth', 'nor', 'not', 'one', 'then']
state 8    words: ['love', 'fair', 'self', 'new', 'art', 'am', 'me', 'her', 'might', 'by']
state 9    words: ['it', 'love', 'which', 'thee', 'me', 'and', 'you', 'of', 'self', 'can']
```

Transition Matrix, 20 states, 15 iters

```
state 0    words: ['of', 'with', 'thee', 'in', 'make', 'heart', 'self', 'show', 'all', 'lov
state 1    words: ['thou', 'and', 'it', 'my', 'time', 'you', 'i', 'is', 'so', 'by']
state 2    words: ['for', 'so', 'but', 'in', 'of', 'thou', 'or', 'i', 'what', 'the']
state 3    words: ['i', 'that', 'no', 'but', 'of', 'they', 'do', 'as', 'thou', 'like']
state 4    words: ['thy', 'i', 'with', 'in', 'than', 'the', 'to', 'that', 'from', 'and']
state 5    words: ['and', 'that', 'is', 'thou', 'shall', 'love', 'so', 'have', 'you', 'ther
state 6    words: ['of', 'me', 'with', 'do', 'the', 'thee', 'for', 'in', 'self', 'is']
state 7    words: ['that', 'i', 'thy', 'am', 'in', 'of', 'what', 'the', 'with', 'it']
state 8    words: ['my', 'of', 'not', 'a', 'to', 'thou', 'i', 'in', 'her', 'be']
state 9    words: ['and', 'that', 'have', 'me', 'which', 'for', 'thee', 'from', 'than', 'as
state 10   words: ['and', 'that', 'the', 'my', 'for', 'i', 'but', 'with', 'thy', 'as']
state 11   words: ['the', 'i', 'love', 'by', 'thy', 'this', 'your', 'to', 'can', 'in']
state 12   words: ['o', 'and', 'when', 'for', 'if', 'who', 'that', 'then', 'which', 'whilst
state 13   words: ['and', 'thee', 'to', 'that', 'my', 'heart', 'will', 'be', 'where', 'have
state 14   words: ['and', 'but', 'when', 'which', 'so', 'nor', 'mine', 'for', 'that', 'why
state 15   words: ['the', 'i', 'it', 'or', 'which', 'then', 'when', 'thou', 'so', 'by']
state 16   words: ['to', 'his', 'my', 'doth', 'their', 'a', 'and', 'mine', 'thine', 'some']
state 17   words: ['in', 'be', 'thou', 'is', 'not', 'all', 'of', 'love', 'self', 'with']
state 18   words: ['to', 'my', 'thy', 'a', 'in', 'your', 'his', 'and', 'their', 'doth']
state 19   words: ['the', 'and', 'thee', 'from', 'for', 'but', 'can', 'nor', 'this', 'thy']
```

Properties of Hidden States

What are some properties of the different hidden states? e.g. Correlation between hidden states and syllable counts, connotations of words, etc. [TODO]

5 Additional Improvements

Rhyme

We incorporated rhyme by picking seven pairs of rhymes, arranging them in the correct order, and using them at the end of the line. The lines generated forward would often not match with the rhymes at the end, so we switched to generating each line backwards from its ending rhyme.

Additional texts: *Hamilton* songs

Shari?

6 Conclusion

Division of work

Shari worked on getting the initial 14-line sonnet, ensuring that lines were 10 syllables with resampling, and incorporating additional source material.

Carly worked on rhyming, seeded and backwards generations of emissions, and interpretation of hidden states.

Discoveries

What are your conclusions/observations about the models you used and the sonnets generated?

Challenges

Concluding Remarks

- hey there