

CS221 Project Proposal

Team: Fangzhou Liu, Amitkumar Gamane, Atticus Geiger

1. Input-output (with examples)

Input: First 12 lines of a Shakespearean sonnet

Output: Computer-generated final rhyming couplet (last two lines) of the same Shakespearean sonnet

Example: See baseline and oracle examples below.

Scope: We will use 152 of Shakespeare's sonnets for training and testing sets.

2. Evaluation metric

Test: Given a set of 5 sonnets, in which 4 are actual sonnets in the full and the fifth has had the last two lines (final couplet) replaced by non-Shakespeare output, ask randomly-selected Stanford engineering students to identify the sonnet with the non-Shakespeare couplet. We will not tell participants that the sonnets were written by Shakespeare to ground participant responses in their judgement of poetic tone rather than prior preconceptions about Shakespeare.

Baseline: Replace the final couplet in a Shakespearean sonnet with output generated by a n-gram pathfinding couplet generator (see example below).

So oft have I invoked thee for my Muse,
And found such fair assistance in my verse
As every alien pen hath got my use
And under thee their poesy disperse.
Thine eyes, that taught the dumb on high to sing
And heavy ignorance aloft to fly,
Have added feathers to the learned's wing
And given grace a double majesty.
Yet be most proud of that which I compile,
Whose influence is thine, and born of thee:
In others' works thou dost but mend the style,
And arts with thy sweet graces graced be;
'Gainst death which he live
Had my heart.

Oracle: Replace the final couplet with a couplet written by a poetry student (see example below).

So oft have I invoked thee for my Muse,
And found such fair assistance in my verse
As every alien pen hath got my use
And under thee their poesy disperse.
Thine eyes, that taught the dumb on high to sing
And heavy ignorance aloft to fly,

Have added feathers to the learned's wing
And given grace a double majesty.
Yet be most proud of that which I compile,
Whose influence is thine, and born of thee:
In others' works thou dost but mend the style,
And arts with thy sweet graces graced be;
Mine hand unclothe thy light hand from dim earth
Mine pen uncloak thy eye and give thee birth.

Gap: 30% difference between rate at which baseline passes test (estimated 0%) and rate at which oracle passes test (estimated 30%)

3. Challenges

Syntax: Capturing syntax of whole sentence (information related to whole path) in individual state -- i.e. designing minimal state information is a challenge

Nature of poetic language: Poetic language often operates on the subversion of expectation with its use of metaphor and unconventional syntax. This may pose problems for the use of frequency n-gram approaches or standard NLTK tools, which have been successfully applied mostly to problems involving non-poetic language.

Themes: Staying “on topic” -- i.e. in line with thematic focus on poem -- creates the dual problems of identifying and then remaining consistent with theme of the first 12 lines of each sonnet. We foresee that the machine learning problem involved in clustering words by theme, especially when words have metaphorical meanings, will be difficult as well

4. Relevant topics

Search problem: We plan to model the process of generating the last couplet will be modelled as a search problem. Given that the final word in each line of the couplet must rhyme, the pair of final rhyming words (one for each line) will be pre-selected as end states. Each line is then generated using n-grams, where every possible word explored is a “state” with a cost associated and future states (based on n-grams containing the past (n-1) sequence of words).

Machine learning: We will make use of feature selection and weight learning to design a cost function for each subsequent n-gram added to the final couplet, capturing information such as syntax in addition to n-gram costs. These costs may be integrated as heuristics.

5. Related work:

- Translation of poetic/literary tone analysis to a language generation problem.
 - [Kao & Jurafsky](#): “A computational analysis of poetic style”
 - [Kao & Jurafsky](#): “A computational analysis of style, affect and imagery in contemporary poetry”
 - [Greene et al](#): “Automatic analysis of rhythmic poetry”
 - [Gervas](#): “An expert system for the composition of formal Spanish poetry”
- [Andrej Karpathy's](#) neural networks Shakespeare project