

Generating Topical and Humorous Poetry

Chetan Bademi chetanbademi@gmail.com

Problem

Given a topic, generate a funny four line poem, with rhyming scheme AABB, where the last line is the punchline.

Previous Work

Ghazvininejad et al[1] generate poems from a topic by generating related words, choosing rhyme words which will be the end of each line, creating an FSA which encodes all valid poems and finding a good path in the FSA using a model trained on song lyrics.

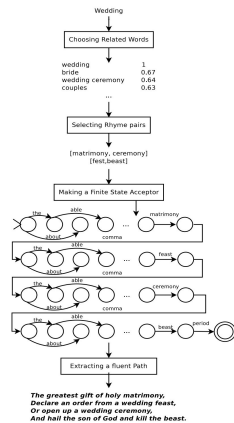


Figure 1: Generating Topical Poetry, Ghazvininejad et al[1]

Weller et al [3] use a transformer neural network architecture, to classify good jokes, which can predict the level of humor for a specific audience to a higher degree than a general audience consensus.

Methods

We used the methods described by Ghazvininejad et al[1], henceforth referred to as “**Baseline**”, with modifications to how rhymes are chosen and the model used to generate the poems.

Adding Humor: Joke Sequence model

Baseline: Encoder-decoder model, which generates four lines, given four rhyme words, trained on song lyrics.

Intuition: A model trained on song lyrics does not understand or cannot generate humor.

Dataset: 200k jokes from r/jokes on Reddit. ~7M words.

Model: 3-layer RNN with 1024 hidden states.

Inference: Constrain by FSA and rhyme words.

Choosing better rhyme words: Rhyme Word Sequence model

Baseline: Pairs of rhyme words are chosen independently, with probability proportional to the product of similarity to topic for two words in the pair. Order of pairs is random.

Intuition: Learn which rhyme words are used together in pairs often and which pairs should come after others, ie, a scoring function for sequence of four rhyme words.

Dataset: From song lyrics dataset, choose all consecutive lines with rhyming scheme AABB and use last words of the lines as one example for the sequence model. ~60k examples (four words each) from ~300k songs.

Model: 1-layer RNN with 128 hidden states.

Inference: Enforce rhyme scheme using FSA. Words at ends of line 2 & 4 should be in top 10 related words. Intuition is that poets/songwriters often setup context with unrelated rhyme words but deliver a punchline with a closely related word if absolutely have to rhyme.

Evaluation

To evaluate our approach, we generate poems for each method for 100 audience suggested topics for the game “Hoedown” from the TV show “Whose Line Is It Anyway?”. The poems generated by professional improvisers is . The language model(LM) score is the perplexity of the poem according to OpenAI GPT model. The good joke probability is using the humor detection classifier by Weller et al[3].

Results

Method	LM Score		Good Joke Probability	
	Mean	Stddev	Mean	Stddev
Baseline	229.41	100.04	24.70%	20.70%
Baseline + Rhyme Sequence Model	127.31	47.23	27.60%	12.00%
Baseline + Joke Generative Model	284.36	99.97	27.20%	14.50%
Baseline + Rhyme Sequence Model + Joke Generative Model	185.96	72.09	28.90%	14.30%
Human Improvisers	46.27	31.52	58.90%	19.70%

Table 1: Results from generating poems for 100 topics.

Baseline	Baseline + Rhyme Word Seq Model	Baseline + Rhyme Word Seq model + Joke Seq Model
I asked my little lady princess caroline! Grabbed her hair and offered me a new valentine! The son of many named elisabeth christine. Had a gift of love and such an old wedding scene.	Another crazy way of making me forget. Life was such a very happy day that you met! The grass is blue and greener on the other side! Looking through the eyes of love and see a new bride.	They love about a moment after you forget! Many years ago and those who haven't met. Or such an older lady on the other side! Thought of me the other day and bought a new bride.

Table 2: Poems from different methods for topic “wedding”³

Discussion

- [1] Using Rhyme Word Sequence Model to choose rhymes improved both language model and good joke scores, irrespective of model used to generate the rest of the poem. The improvement is particularly significant when the song lyrics model is used, because the rhyme word sequence model is trained on the same dataset and chooses rhyme words, which song lyrics model knows most about.
- [2] Using the Joke Sequence Model for generating the poem makes the poems less sensible but more funny.
- [3] None of the methods come close to human improvisers.

Help evaluate this project: bit.ly/funnypoemdemos

References

- [1] Ghazvininejad, M.; Shi, X.; Choi, Y.; and Knight, K. 2016. Generating topical poetry. In EMNLP, 1183–1191.
- [2] Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an interactive poetry generation system. Proceedings of 55th ACL, pages 43–48
- [3] Orion Weller, Kevin Seppi. Humor Detection: A Transformer Gets the Last Laugh. arXiv:1909.00252. EMNLP 2019.