

Joke's On You: An Exercise in Joke Generation

CONNOR FORD AND GABE MAGEE

Pomona College

connor.ford@pomona.edu, gabe.magee@pomona.edu

May 7, 2019

ABSTRACT

Joke generation is a difficult task for humans and machines alike. It necessitates context, timing, and wordplay to elicit a laugh. We consider a subclass of 'knock-knock' jokes to simplify the generative and use a variety of straight-forward algorithms to create a small number of jokes with origins from movie scripts and the web.

I. INTRODUCTION

There have been many successful approaches to joke generation.¹ These jokes typically follow a certain rigid structure such as call-and-response. But others have trained more intelligent models on large corpuses of data scraped from Reddit or Twitter. These have less associated structure and generally see more mixed results. We want to consider a less common joke-type in the current literature: the 'knock-knock' joke. Considering this joke type has a couple main advantages. (i) It follows a rigid structure. To illustrate the format, we annotate the following classic 'knock-knock' joke (not generated).

A: Knock knock.

B: Who's there?

A: Cash. [*Token*]

B: Cash who? [*Token + who \approx search word*]

A: No thanks, I'll have the peanuts [*Play on search word*]

(ii) There is a discrete set of 'knock-knock' joke subtypes². One of which, as Taylor identifies, is word play on the *token*. We pursue this specific type because it allows us to take a more straightforward, algorithmic approach to generation. The algorithm looks like:

- (a) Generate *Token* from dictionary/corpus by identifying *search words* that end in the 'who'/'ew' sound and extracting the root of these words.
- (b) Use corpus to generate responses based on the *search word*.

II. METHODS

i. Generating Tokens

We used a corpus of $\sim 300k$ words and found words ending in the following substrings: ["who", "ew", "ewe", "ooo", "ooh", "oo", "hue", "ue", "eau", "eww", "hu"]. We cross-referenced the *tokens* with another English language dictionary. This cross-referencing only generates a list of 9 potential english words. To extend this, while sacrificing a bit of quality, we cross-referenced tokens with a pronunciation dataset based on a CMU pronunciation dictionary with words ending in ["UW0", "UW1", "UW2"]. This

¹Cai, J., and Ehrhardt, N. (2013). Is This A Joke?. Yoshida K., et al. (2018). Neural Joking Machine: Humorous image captioning. Mihalcea, R. and Strapparava, C. (2006). Learning to Laugh (Automatically): Computational Models for Humor Recognition.

²Taylor, J. (2004). Computationally recognizing word-play in jokes.

outputs a list of about 68 english *search words* but the quality of the tokens took a hit. For example, it generates tokens such as 'sh' corresponding to 'shew'.

ii. Generating Responses

We decided to generate a response to a joke one of two ways:

- (i) Using a bigram approach learned from a Wikipedia dataset. This meant stringing together similar words to artificially construct a response.
- (ii) Using a 'sentence following' approach from a movie script dataset. This meant finding the occurrence of a word in the movie script data set then returning the next sentence in the dialogue. This dataset is nice because movie scripts allow for more conversational flow than most other corpuses do. It also includes various colloquialisms making it well-suited for jokes.

III. RESULTS

token	search word	response
Bigrams		
cash	cashew	cashew apple ii cd of the target for championship at 124.
Sentence Following (1)		
fug	fugue	Fuck that, maybe you're my hallucination.
resid	residue	Putty? On both doors?
mild	mildew	And is that it?
rev	revue	The job's gonna cost you a hundred bucks.

Sentence Following (2)

wa	wahoo	That's my point!
gl	glue	What's it to you?
curf	curfew	You think he could still be in town?

IV. DISCUSSION

i. Why Human Evaluation

We decided to do human evaluation because of the nature of the results. It's really hard to evaluate a joke quantitatively using machine-based methods, due to the lack of models about them. This leaves only a few options, among them chiefly is human evaluation. A score generated from collective human reactions to jokes generated may be a good way to evaluate our models.

ii. Evaluation Methods

To test our results, we would have a survey that evaluates for different methods for producing jokes. It would include a few generated by each methods along with a place to score the joke as a whole for its coherence and humor.

iii. Evaluation Results

The survey has not been finished... to be continued.

REFERENCES

- [Yoshida, Kota et al, 2018] Kota Yoshida and Munetaka Minoguchi and Kenichiro Wani and Akio Nakamura and Hirokatsu Kataoka. (2018). Neural Joking Machine: Humorous image captioning. *CoRR*, abs/1805.11850.
- [Cai, J., and Ehrhardt, N., 2013] Cai, J., and Ehrhardt, N. (2013). Is This A Joke?.
- [Mihalcea, R. and Strapparava, C, 2006] Mihalcea, R. and Strapparava, C. (2006), Learning to Laugh (Automatically): Computational Models for Humor Recognition.

Computational Intelligence, 22: 126-142.
doi:10.1111/j.1467-8640.2006.00278.x

[Taylor, 2004] Taylor, J. (2006), Computationally recognizing wordplay in jokes *Cognitive Science*