An Ernie-st attempt at humor classification and rating

Amy Wang; Zheng Yan

Introduction

In the past decade, artificial intelligence has excelled in many areas in NLP.

Humor, despite its prevalence and importance in human conversation, is not one of these areas.

We would like to create a system that accomplishes two tasks:
(1)Classifies an utterance as a pun, or not a pun;
(2)Gives a rating to the funniness of the pun.

In doing so, we hope to contribute to the broader progress of bettering AI-human interactions, and present techniques that can make AI companions appear friendlier.

We build the first natural language processing system that implements the cognitive science-based model presented by Kao et al. in 2015. This model was used to effectively predict the funniness of an ambiguous sentence, based on two metrics: **ambiguity** and **distinctiveness**, both originally estimated via human crowdsourcing.

Approach

"Throwing acid is wrong, in some peoples' eyes"

"Throwing acid is wrong, according to some people"

"Throwing acid is wrong, at eyes of some people"

Is this a pun? Let's math it out!

P("Throwing acid is wrong" | "according to some people") * P("according to")
~=
P("Throwing acid is wrong" | "at eyes of some people") * P("at eyes")

High ambiguity: High entropy of P(meaning | sentence)!

P("acid" | "at eyes") > P("acid" | "according to")
P("Throwing" | "at eyes") > P("Throwing" | "according to")
P("is wrong" | "according to") > P("is wrong" | "at eyes")

High distinctiveness: High KL divergence of P(words in sentence | meaning)!

Figure 1 Approach

Background

The model defined by Kao et al. is as follows:

$$Amb(M) = -\sum_{k \in \{a,b\}} P(m_k|\vec{w}) \log P(m_k|\vec{w})$$

$$Dist(F_a, F_b) = \sum_i \left(\ln\left(\frac{F_a(i)}{F_b(i)}\right) F_a(i) + \ln\left(\frac{F_b(i)}{F_a(i)}\right) F_b(i)\right)$$

- Intuitively, **ambiguity** measures the relative fluency of the sentence when either interpretation of the punned word is used.

- Intuitively, **distinctiveness** is the ability of the sentence to draw attention to ambiguity. For instance, "Look at that hare" is technically ambiguous, but the phrase does little to point it out.

These ideas are derived from the theory that lexicographical **priming is necessary to facilitate the disambiguation of sentences**, a process which leads to humor when expectations are subverted.

Figure 2 Model performance

Methods

- **Metrics**: We used mean squared error for the rating task and accuracy for the classification task.

- **Baseline**: We used an end-to-end LSTM baseline to predict scores and classification directly using GloVe word vectors.

- **Ambiguity**: We used a BERT with a masked language model to estimate the distributions of words given sentence meanings. After filtering for noise, we then take the entropy of these distributions.

- **Distinctiveness**: We attempted 2 approaches: 1) using the distribution of word predictions 2) summing the attention paid the ambiguous word/phrase by all other words in the sentence, averaged among all attention heads and layers. We then take the K-L divergence of this.

- **Final output**: We trained a decision tree ensemble model and a multilayer perceptron on these two values for a final classification and rating.

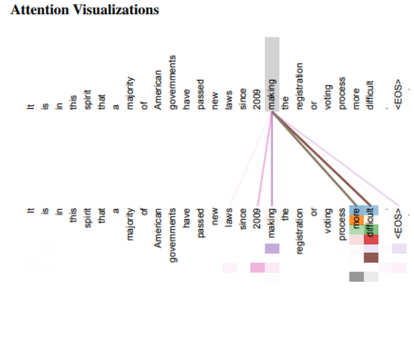

Figure 3 BERT's attention mechanism (from [2])

Data

We used three datasets:

Reddit dataset: 400 puns hand-scraped from Reddit. Includes the ambiguous words/interpretations.

iWeb subset: 1600 non-jokes scraped from various websites. Sentences from this dataset with the same ambiguous words as in jokes were mixed with joke data for the classification task.

Kao et. al. dataset: 100 puns and 300 non-puns, labeled with a crowdsourced "funniness" rating, as well as ambiguous words/interpretations.

Results and Analysis

Overall, our final model was unable to outperform our LSTM encoder baseline (Figure 2).

First, we analyzed BERT's efficacy for our task. In addition to low perplexity, we found that BERT's estimates of word likelihood were often empirically correct:
"Magician" is more likely given "hare" than "hair" (4.19e-05 vs. 3.91e-05).
"Mad" is more likely given "hair" than "hare" (6.07e-05 vs. 5.00e-05)

However, we do see a significant amount of errors. Unfortunately, many words used in puns are easy for humans to make semantic connections for, but do not co-occur together contextually.

With our second approach, it also did not appear that BERT's attentional model was able to capture relatedness between words. The idea that BERT's attentional layers, across 12 layers and averaging over attention heads, could be used to identify semantic similarities is experimental. We are not too surprised that this had a negative outcome.

Conclusions

We created a neural implementation of a method for humor classification and rating based off of cognitive science research by Kao et. al. Unfortunately, the model did more poorly than a baseline LSTM to predict the scores directly from sequences of word vectors.

Mostly, low accuracy stemmed from the inability of language models to semantically connect words that, while loosely related, do not co-occur. More successful computational models in the future might be based on techniques that more accurately map relationships between words, such as solutions founded on principles of embeddings or dimensionality.

References

[1] Kao, J. T., Levy, R., \& Goodman, N. D. (2016). A computational model of linguistic humor in puns. Cognitive science, 40(5), 1270-1285.
[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems(pp. 5998-6008).