Humor Response Generation: Machines Can Learn To Be Funny Too

Tyler Ryu, Tsung-Chin Han
UC Berkeley School of Information
{tylerkryu, tsung-chin.han}@berkeley.edu

Abstract

Humor, in the natural language, presents quite a dilemma when it comes to computational natural language processing, as the incorporation incur technical challenges that far exceed the complexity of processing and generating the primary contexts. It is the thin polish required to fully mimic the subtle nuances of any natural languages. The fact that the meaning behind humor often appears to be special or subtle compared to a normal conversational language. That makes machine pretty challenge to understand and provide values when conducting this kind of communications. In the case we usually could see responses which are calculated, monoexpressive, and devoid of any emotions or humor to a point that even people with such speech traits are often regard it as "robotic". In this short study, we explore the popular natural language processing with deep learning models in question-response generation field. We experimented those models on the humorous source from the open internet.

1 Introduction

The advent of the transformer natural language processing model has brought text generation to a new level. Model such as BERT (Devlin et al., 2019) achieving the state-of-the-art performance in natural language understanding tasks. GPT-2, (Radford et al., 2019) capable of generating extended amounts of coherent paragraphs, is also achieving advanced performance in natural language modeling tasks. Both of them have ability to generate well-coherent and natural articles. The GPT-2 pre-trained model was introduced with a staggered release by OpenAI, because of the fear that the model would be used for malicious intention, such as fake news generations or document impersonation activities. Accordingly, they were cautious of the releasing until the full pre-trained model with 1.5 billion parameters has introduced.

Humor generation has been tackled far less than other aspects in natural languages. This may due to researchers argue that the domain is beyond verbal communications (Morkes et al., 1999). With the recent breakthroughs, we started to see some studies in the field, such as pun generation by Dybala (Pawel Dybala et al., 2008), where they attempted to incorporate humor into natural language using paronomasia, exploiting word sound-meaning into existing texts. From the work of humor generation by Ren & Yang (He Ren and Quan Yang, 2017), we can understand the implementation of LSTM in generating new sentences that are considered humorous. The downsides of those previous attempts are that humor can only be incorporated as an add-on (replacement of words with puns) or a single statement joke (monologues). With those background in mind, we are curious of applying a transformer-based model in the domain, because we vision the model has capability of adapting a given statement and thus providing a contextually appropriate response.

In this experiment, we combine a fine-tuned GPT-2 with a Bidirectional LSTM for creating effective humor responses and sanitizations for the given questions or statements.

2 Data

The dataset used for our experiment is from the popular social platform Reddit under the subreddit "r/Jokes" section. The dataset we called "Reddit Jokes" or "r/Jokes" in our study contains a total of 194,553 instances of jokes. In our work, due to the nature of the multi-model implementation, a data pipeline is required to streamline the input-output processes. Meanwhile, the two separate workstreams of models themselves require separate training and fine tuning using altered ver-

sions of the same original source dataset.

2.1 Humor Generation Data

The Reddit jokes dataset contains roughly 195,000 instances of jokes, which constitutes paired title-body data points. The title of the joke contains either a question or a statement that is generally left unfinished. This was complete by the body contents. Minimal preprocessing is required for the model training sets. Our desire input is to retain the conversational tones associated with posts uploaded by Reddit users that are closely resemble common conversational speech patterns. Preprocessing work generally consists of adding prefixes to the title and body sequences (e.g. "<|startoftext|>", "<|endoftext|>" to allow for delimiting the outputs of the fine-tuned GPT-2 model).

2.2 Humor Recognition Data

In terms of the downstream task, in order to evaluate the performance on humor recognition, we require our training data from the Reddit jokes dataset to include both positive (humorous) and negative (non-humorous) samples, in which we differentiate these binary features based on the post score displayed from the original Reddit jokes data. Similar to our preprocessing of the upstream task, we concatenate the title and body sequences with prefixes "title:" and "body:" to form our expected input to the downstream language model. Table 1 shows the outline statistics of the training dataset. The form of the training data in the humor recognition task is then well in line with our testing data, the outputs of generated sequences from the humor generation task.

Dataset	**# Positive**	**# Negative**	**Language**
r/Jokes	113,645	80,908	English

Table 1: Outline Statistics of the Training Data

3 Approach

In this section, we describe our experiment framework of Humor Auto-Generation and apply it into the Humor Recognition task. Jokes are generated by a language model GPT-2 and then subsequently used as inputs to the LSTM architecture for a text classification problem.

The particular implementation of the two language models was chosen based on their ease of implementation and performance. However, it is to be noted that the similar results are achievable with BERT, TransformerXL (Dai et al., 2019) and binary classification models, although BERT may present a divergence from GPT-2 due to the non-autoregressive nature of encoder stacks used in BERT compared to GPT-2s decoder stacks.

3.1 GPT-2 Fine Tuning

In our experiment, GPT-2 of any parameter size can be used, with higher parameter size being preferable, but a careful balance of performance and size is required given the available resources (8 vCPUs, Intel Xeon 2.7 GHz, 1 NVIDIA Tesla K80, 30 GB Memory). The fact that multiple variants of GPT-2 models are available, we chose the 345 million parameter model to be our baseline, as the responses from this model closely resemble the natural language for the targeted humorous responses. Parameters for the model itself include a top-k truncation of 40 (40% top logits, accounts for the likelihood of prediction), and a temperature value of 0.85 (logit scaling before softmax, accounts for the diversity of predictions). These two combinations yield the best results. As described in Table 2, a low top-k truncation resulted in looping sentences whereas low temperatures and extreme high temperatures resulted in either mundane text (more in line with the baseline model response) or incoherent text (from high randomness).

Parameter Variant	Sample Output
Low Top-K (k=1)	"There was a man and there was a woman and there was a man and there was a woman… and there was a man and there was a woman."
Low Temperature (t=0)	"" (empty string)
High Temperature (t=100)	"Chicken road KFC back consume highway intersection"

Table 2: Sample responses from the fine-tuned GPT-2 model (345M) producing ineffective outputs given different extreme parameters

With 3,538 steps (epochs) in our work, the final model achieved an average loss of 1.90 (starting at 2.90). The total number of steps was chosen based on an estimation of 3,000 steps with eight hours of computation time.

3.2 Bi-Directional LSTM

Long Short Term Memory networks (LSTM) was proposed by (Hochreiter and Schmidhuber, 1997) and were refined and popularized by Felix Gers et al.. LSTM shows well on a large variety of problems including text classification tasks. LSTM network architecture is explicitly de-signed to avoid the long-term dependency issues. The LSTM maintains a separate memory cell inside the network that updates and exposes the content only when deemed necessary.

In our experiment, we regard the humor recognition downstream task as a text classification problem. With the ease of implementation and the performance compared to other network architectures, Figure 1 exhibits our chosen model details. We converted tokenized input sentences with word vectors by GloVe embeddings (Pennington et al., 2014), where we trained our model on top of the 6B tokens, 400K vocab, and 300 vectors of Wikipedia 2014 + Gigaword 5 version. Following the embedding layer is a Bidirectional LSTM network with different regularizations and dense layers thereafter to flatten the outputs.

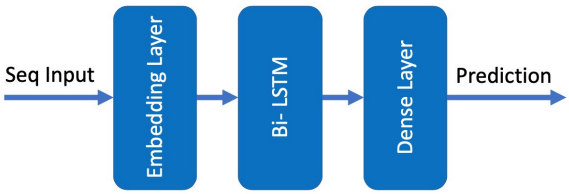

Figure 1: Network Architecture

3.3 Experimental Framework

The framework of the two language models in our experiments involves two aspects: the inputs to the GPT-2 model (the title sequence with the delimiter notation without the end of text notation) for sample joke generations, and the subsequent outputs that are fed into the Bidirectional LSTM model. In the end, we attempt to understand the final outputs as a binary text classification problem — humorous or non-humorous. Our two networks of language models are hosted on separate environments in order to have asynchronous responses to remediate the slow processing time for the GPT-2 model.

3.4 Human Evaluation

As part of the nature of humor generation, using conventional means of benchmarking would likely produce inaccurate results. The algorithms such as BLEU (Papineni et al., 2001) is fairly useful for machine translation, but BLUE heavily favors word pattern similarities (n-grams) and does not take into consideration the actual meaning behind the words. Humorous responses, on the other hand, often present complete new or contextually divergent words, such as antonyms. Without proper understanding of the contents, it is pretty difficult to measure and benchmark the outputs of humorous responses. As such, in our experiment, we introduced a human evaluation scheme as our benchmark for the downstream evaluation methods, even though the scheme does not set the gold standard on par with humans.

In the human evaluation scheme, a survey of hundred sample titles of jokes were selected. For each of the title, we generated three times from the fine-tuned GPT-2 to have three different response pairs. Therefore, a total of three-hundred title and body pairings were selected at random and presented to eight test participants. The participants were then asked to select the funniest pairs in one of the three choices. Finally, we compare the human selection results with the Bi-LSTM classifier.

4 Result

We present results following our experimental framework we described in section 4. In the Humor Recognition task, we validated the training performance of different network structures. For the testing performance, we leveraged the Bi-LSTM model to classify the auto-generated joke outputs from our fine-tune GPT-2 model. We collected the machine results and compare that with the human evaluation benchmark to better understand the successfulness of the auto-generated jokes.

Table 3 describes the training results of the Bi-directional LSTM model on the Reddit jokes dataset (r/Jokes). We chose a dropout rate at 0.1 and recurrent dropout rate at 0.3 respectively.

Dataset	Accuracy	Loss	F1
r/Jokes	0.62	0.65	0.72

Table 3: Training Results of Reddit jokes

As described in section 3, we trained the LSTM network on the preprocessed Reddit Jokes data (r/Jokes). We can see that in terms of the F1 measure, we achieved the score at 0.72. Together with the training and validation processes shown in Figure 2, Figure 3, and Figure 4, it shows that our implementation of the Bidirectional LSTM model can learn the humorous meaning and structure embedded in our sequence of inputs but with limited ability to rely on the classifier to differentiate our targeted contents.

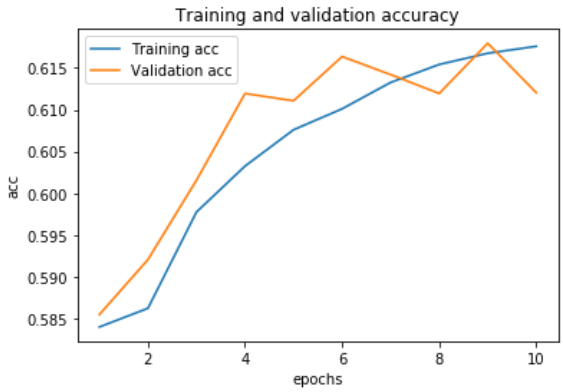

Figure 2: Training and Validation Accuracy

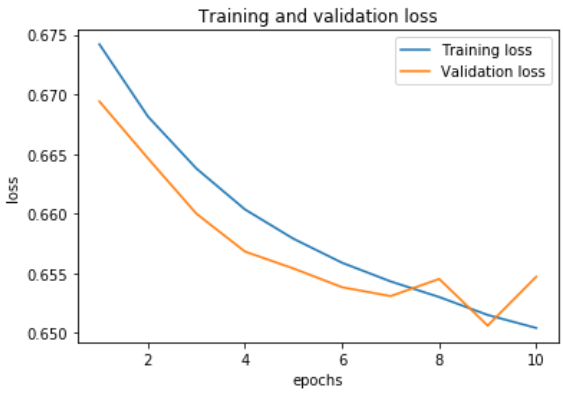

Figure 3: Training and Validation Loss

Figure 4: Training and Validation F1 Score

Despite the fact that our implemented Bi-directional LSTM model may not be a strong classifier for our downstream task, we tested three-hundred joke samples, in which each title of joke, we generated three times to have three different responses from the fine-tune GPT-2. Next, we have the Bi-LSTM to classify those joke samples and output the predictions and scores (probabilities) associated with it.

As shown in Table 4, the results from the human evaluation scheme, more than half of the people from eight participants agreed on the question-answering pairings. That being said, we were able to see there is a correlation with the human consensus from our humor generation task. However, compared to our downstream classifier, it only exhibited 21% accuracy out of 47 pairs.

# Samples	# Consensus	LSTM Consensus	%Matched
100	47	10	21.27

Table 4: Human Evaluation Results

5 Conclusion

In this study, we have extended the technique of common-sense question-answering work on humor generation in which we create a humor response to any given natural language question and statement. We proposed a fine-tune GPT-2 model that can learn to effectively generate humorous contents based on the dataset that came from Reddit under the sub-reddit "r/Jokes" section. Regarding whether the performance of each of the auto-generated contents is humorous or non-humorous, we also introduced a deep learning Bidirectional LSTM architecture that can help us automatically conduct the downstream humor recognition work. Separately, we integrated a human evaluation scheme that allows us to better judge the performance of our auto-generated pairs as well as compare the results from our Bi-LSTM machine counterparts. Although the Bi-LSTM model relieves the required human intervention of selection linguistic features for humor recognition tasks, the performance of the model itself shows the classifier has limited ability in our experiment. However, it is worth mentioning that the outcome of the human evaluation benchmark was relatively positive. More than half of the people from our participants agreed on the question-answering pairings, which means we were able to see that the human consensus is much in line with our auto-generated pairs from the humor generation tasks.

For the future study, in terms of auto-generated pairs, we would leverage bigger versions of GPT-2 (1.5B) to understand if we can achieve higher human consensus. For our downstream task, we look to improve the classification model by collecting and training on more joke datasets including different length of pairs, jokes types, data size, and different languages. We also look to introduce a more advanced network architecture with deep learning to see if we can later construct a stronger classifier for humor recognitions. Together with a much rigorous human evaluation scheme we seek to integrate into the work, we also would like to study the ethical implications of biases that are accumulated during pre-training.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* arXiv:1810.04805 [cs], May. arXiv: 1810.04805.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. *Language Models are Unsupervised Multitask Learners. :24.*

J. Morkes, H. K. Kernal, C. Nass. 1999. *Effects of Humor in Task-Oriented Human Computer Interaction and Computer-Mediated Communication: A direct Test of SRCT Theory. Human-Computer Interaction Volume 14, 1999.*

P.Dybala, M. Ptaszynski, S.Higuchi, R.Rzepka, K. Araki. 2008. *Humor Previals! - Implementing a Joke Generator into a Conversational System. AI 2008: Advances in Artificial Intelligence, 2008*

H. Ren, Q. Yang. Final Project Reports of Course CS224n, 2017. *Neural Joke Generation.*

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. *Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. arXiv:1901.02860 [cs, stat], June.* arXiv: 1901.02860.

S. Hochreiter, J. Schmidhuber. 1997. *Long Short-Term Memory, Neural Computation* 9(8):1735-1780, 1997

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. *Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar.* Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. *BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* - ACL '02, page 311, Philadelphia, Pennsylvania. Association for Computational Linguistics.