LLMs in Humor Generation

Jason Nguyen
Department of Computer Science, Rice University
jn61@rice.edu

Abstract

We explore the potential of large language models (LLMs) in understanding and producing humorous content. Humor comprehension presents a unique challenge for language models due to its subjective nature, requiring not only semantic reasoning but also consistently updated contextual and linguistic subtleties. We fine-tune three models, Google Flan T5, Mistral, and BART on a filtered dataset of highly up-voted and short jokes from the popular Reddit forum r/Jokes. Our aim is to assess a model's ability to generate punchlines that either align with the setup provided in the jokes or provide reasonable comedic effect. We utilize manual evaluation from several individuals to assess which model performs best and also whether or not LLMs can generalize well to comedic text generation. Through this investigation, we show the extent to which LLMs can capture and replicate subjective literature through humor, and present reasonable performances on this task.

1. Introduction

Creating humorous content is a difficult task even for humans, as the standards for what is considered funny can vary significantly between individuals. Typical jokes tend to involve witty play on words, commentary about current or historical events, or widely accepted or acknowledged standards from social norms and stereotypes. Furthermore, jokes assume that the audience has at minimum some knowledge about the subject matter at hand and occasionally even the speaker.

It follows that understanding these unspoken nuances within comedy is also a challenge for computers. Recognizing and generating humor involves the computer comprehending significant context, but also the subtle manner in which the joke is delivered. The saying "delivery is everything" holds very much true for comedy. However, the task of incorporating speech factors such as tone, speed, and accent into text data is an obstacle itself. The problem only becomes more extreme when accounting for the fact that senses of humor tend to evolve and expand over time.

We utilize a scraped dataset of posts [4] from the popular Reddit forum `r/jokes`. The dataset is arranged such that the title of the post is the joke question and the body of the post is the punchline or followup to the question. We filtered for short, single sentence punchline posts with a karma score greater than 20. Reddit karma score is the number of likes minus the number of dislikes on a post; a larger karma score implies higher positive engagement. We deem that a karma score higher than 20 indicates a joke that people found sufficient funny and a good threshold to filter out our dataset with. We then define short posts as those having less than 256 characters and being a few sentences long at most. After filtering, we had 12503 pairs of titles and bodies for our training set and 695 pairs for our validation set.

2. Related Work

There has been sparse work on the topic of LLMs and humor. Generally, such works account for humor recognition as an extension of human language understanding in computers. For works with a greater focus on humor, there are works such as [3] focus on evaluating different prompts and the associated outputs from ChatGPT while works such as [1] focus on different methods for classifying whether a text is humor or not., While we also utilize text-to-text transformers, Google's Text-To-Text Transfer Transformer (Flan T5) and Bidirectional and Auto-Regressive Transformer (BART), we instead aim to tune them for comedic generation instead of evaluation or classification. Furthermore, we have not seen any other works involving LLM tuning with posts from `r/jokes`.

3. Models

We individually fine tune three different models, Google Flan T5, Mistral, and BART, to get the best performance possible for each model.

T5 is an encoder-decoder model that was pre-trained to be flexible on various unsupervised and supervised tasks in a text-to-text format. T5 was chosen due to being one of the most versatile pretrained LLMs. There are several variants of T5 available with more parameters (more parameters in-

Title	**Body**	**Score**
My girlfriend burned our Hawaiian pizza today...	I should have told her to put the oven on aloha setting.	1218
Since it started snowing, all my grandma has done is stare through the window.	If it gets any worse I'll need to let her back in.	4215
My therapist says I'm paranoid	He didn't *actually* say that but I know he was thinking it.	2549
90s kids won't get this . . .	Social Security benefits.	14604

Table 1. Examples of jokes with high amounts of karma

dicate more memory but generally better performance), but we utilize the base model for memory efficiency.

Mistral is a decoder only model with variants designed for instruction supervised fine-tuning (SFT). We tuned Mistral-7B-Instruct-v0.1, a version designed for the use of chatbots and conversations. Since Reddit posts are in a internet forum with a title and accompanying body to every post, we hypothesized that this format was suitable for the chatbot format. We suspect the informal nature of Reddit posts would match well with the casual responses of most chatbots.

BART is a denoising autoencoder meant for sequence to sequence (seq2seq) tasks such as text summarization. BART is known to perform well for text generation and comprehension tasks, so we also wished to see its capabilities in generating and understanding humor.

4. Experiments

We utilize parameter-efficient fine tuning (PEFT) when tuning each model. PEFT allows us to tune only a small percentage of extra model parameters and freeze the rest. This greatly improves computational and memory cost while not compromising on model performance. Each model was manually tuned with its own Low-Rank Adaptation of Large Language Models (LoRA) configuration [2].

LoRA is the most general PEFT configuration that inserts and trains a small number of new weights instead of the model weights; this is much faster, more memory-efficient and results in a smaller model to store and load. Furthermore, instead of calculating a larger weights updating matrix to generate outputs, it instead approximates it by calculating two smaller matrices. The product of these smaller matrices are shown to tune the original pretrained weights of the model just as well as a full-size weights updating matrix.. We then discuss the tuning specifics for each model.

LoRA configurations have several parameters, which we individually tested for each model. Task type designates the desired function of the model, such as text sequence to sequence, classification, and prompt tuning. Different task types will have a different number of parameters to tune. Thus, it's important to pick the appropriate task to avoid adding too many unnecessary weights. LoraAlpha is the scaling factor of the configuration, where higher values can increase the range of model adaption and generalization. r refers to the size of LoRA matrices used to adapt the model; higher values of r make the model better at learning but greatly increase the number of parameters needed to be trained and subsequently the memory usage. The final values used for each model are in Table 3.

Model	Trainable	All Params	Trainable %
T5	3,538,944	251,116,800	1.40928
Mistral	23,068,672	7,264,800,768	0.31754
BART	884,736	140,305,152	0.63057

Table 2. LoRA allows the training of a small percentage of the total parameters.

	T5	Mistral	BART
LoraAlpha	64	16	32
r	32	8	16
Drop Out	0.05	0.05	0.1
Gradient Accumulation Steps	2	4	16
Learning Rate	1e-3	2e-4	1e-5
Batch Size	8	1	4
Weight Decay	1e-2	1e-3	1e-2

Table 3. Best observed tuning parameters for each model

For T5, we utilized the title of the posts as the input ids and the body of the posts are desired targets of the model. The prompt ""Finish the Joke" was added to the beginning of each input id to guide T5 to generate humorous responses. Other prompts such as "Generate a funny punchline" and "Respond in a comedic manner" were tested but manual evaluation of generated punchlines deemed them not as effective. We specified the task type for T5 as Seq2Seq to establish a relationship between the joke starter and the punchline. Seq2Seq models are known to be well suited for tasks revolving around generating new sentences depending on a given input.

Mistral was the largest model tuned and required the use of the bitsandbytes library to further reduce memory usage. bitsandbytes allows the use of 4-bit quantization in normalized float 4 (nf4) format to further compress Mistral. Storing in nf4 compromises some precision in comparison to bfloat16 and int8 formats but significantly saves on memory.

Only when doing matrix multiplication computations do we convert back to bfloat16 for faster training. The task type used was CASUAL LM, as this is typically the standard when tuning Mistral. We utilized instruction fine-tuning, where the post title was labeled as the user and the post body was labeled as the generated post of the computer assistant. The prompt that worked best was ”You are a bot that responds to jokes with comedic punchlines.”

BART was tuned in a similar manual to T5, but performed better with task type CASUAL LM instead of Seq2Seq. CASUAL LM utilizes more parameters than T5, which most likely impacted tuning as BART is much smaller than T5.

5. Results

We display some of the inputted joke headers, their actual punchlines, and the responses generated by each model.

Header: Why did the old lady fall down the watering hole?

Punchline: She couldn’t see that well.

T5: I’m not sure her body didn’t fall because of some poisonous scum.

Mistral: It was well over her head.

BART: It’s a lot of fun.

Header: How high are you?

Punchline: Officer, it’s suppose to be ”Hi, how are you?”

T5: Anywhere near 90 meters!

Mistral: I’m 420 feet above ground.

BART: It’s a total waste of time.

Header: Why did the chicken cross the road?

Punchline: To get to the other side.

T5: Because it was the best pig I’ve ever had.

Mistral: To get to the other side.

BART: I’m not a big fan of them.

Header: Why does the homeless man drink coffee?

Punchline: He had no proper tea..

T5: Because he has no money.

Mistral: It’s the only hot beverage available.

BART: I can’t believe it.

We opted not to use validation metrics such as ROGUE and BLEU. Manual evaluation of outputs showed that high ROGUE and BLEU scores would occur when the model returned the inputted text as is instead of creating new text. Another common issue of using ROGUE and BLEU was that the models would repeat responses even for different inputs, signifying that the models are failing to generalize. This issue was especially notorious with BART, where even with very low training loss and high scores it would not generalize or create new text. We instead took the manual evaluation of 5 other uninvolved individuals to access our model.

We generated 30 punchlines for each model and had people pick between A, B, or C (Mistral, T5, BART) for whichever punchline they thought was best. We assumed each model was equally capable of generating comedic content, and counted which answer appeared most frequently between each person.

Out of a total of 150 choices made, 102 choose Mistral, 31 choose T5, and 17 choose BART. Our results reflect that the more parameters a model has, the better it generalizes to a specified task. Furthermore, many of Mistral’s response

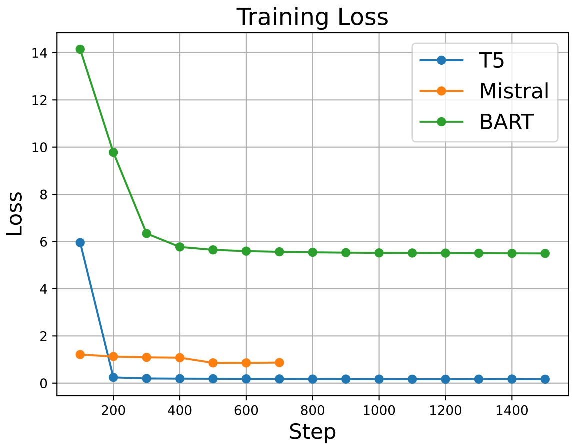

Figure 1. Training loss for each model; Mistral had less than 800 steps due to the batch size and gradient accumulation steps used

were well align with the inputted joke header, even if it was not the same response.

For instance, the header ”How high are you?” would literally refer to height, but Mistral is able to understand that it refers to the high based off marijuana use and response ”I'm 420 feet above ground.” This plays well into both the literal and humorous context of the header.

T5 would most often overfit on the training data and repeat punchlines from the training set, regardless of how different the header was. Since T5 is so flexible, it was uncertain which task type and how many parameters were best to tune T5 with. We choose the version that gave the most general and diverse answers when outputs were generated with a temperature value of 0.97.

BART proved to be the most difficult to tune, as not only would it fail to learn anything from our training set, it would often repeat responses. This is mostly due to the low number of parameters tuned but also the fact that BART works best with text summarization. Since our data was preprocessed to be shorter punchlines, it most likely failed to properly learn any of the proper context. Most of BART's response won votes due to being ridiculous and how out of context it was in response to the header. Our best version still had significant training loss, but at least was able to generate some new responses.

6. Discussion

Recently, there has been the release of Mixtral-7x22B-Instruct and Meta-Llama-3 which have even more parameters available to train. These newer models may prove even more effective at humor comprehension and generation. Furthermore, we only trained on short jokes with direct punchlines. However, many comedic setups can be several paragraphs long and may yield more unique results. While the Reddit dataset was also very diverse with thousands of different jokes, Reddit as a social media forum tends to appeal to the younger demographic. Hence, the humor generated by our work may not appeal to older individuals or those not often on social media.

7. Hyperlinks

Youtube link to presentation
GitHub link to source code and code references
Google Drive with all fine-tune models
Slides

References

[1] X. Guo, H. Yu, B. Li, H. Wang, P. Xing, S. Feng, Z. Nie, and C. Miao. Federated learning for personalized humor recognition. *ACM Trans. Intell. Syst. Technol.*, 13(4), 2022.

[2] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021.

[3] S. Jentzsch and K. Kersting. Chatgpt is fun, but it is not funny! humor is still challenging large language models. 2023.

[4] T. Pungas. A dataset of english plaintext jokes., 2017.