From NLP to LOL: A Witty Reddit Bot

Chittaranjan Velambur Rajan
chitt@umich.edu

Abstract

1 Introduction

The primary goal of this project is to create a bot that can post a top level comment on an arbitrary Reddit post. The intent is to have these comments be witty and humorous, as is usually the case with most Reddit posts. This work is intended as an exploratory endeavor in understanding humour on social media, and building a language model to reflect that. Many others have attempted to tackle similar problems, as is described in the Related Work section, however their applications are more serious in nature while this project aims to be jovial. However, the approach this project takes is similar in that it employs Sequence-to-Sequence transformer models to generate the text that forms the comment. The models perform relatively well, as shown by their perplexity scores discussed in the results. The results also show that zero-shot prompt models performed significantly better than fine-tuned models, and potential reasons for this are discussed in the later sections of the paper.

2 Data

There is a Python Reddit API Wrapper (PRAW) which uses the Pushshift API built by Baumgartner et al. (2020) to interact with Reddit content. Using this API wrapper, relevant data was collected for 2875 top-level comments to serve as the training data. Further, 336 top-level comments were used for evaluating the model.

Each row of the data has the following information:

```
{
"submission_id": Unique Reddit ID
"submission_text": Body of the post
"title":
"permalink": /r/Showerthoughts/...
"comment_scores": [56, 34,109, 1022, ...],
"comment_ids": ['eg2gvmq', 'eg2b8xw'...]
"comment_texts": Body of each comment
}
```

This structure was cleaned to have each row of data have the title and body of the post, along with the top-level comment associated with it.

3 Related Work

Writing an NLP based social media bot is not the most novel idea, and has been attempted before in a variety of contexts.

In a much more serious endeavor than is intended here, Sager et al. (2021) attempt to tackle misinformation on Reddit. Their model attempts to identify misinformation around the topics of essential oils in Reddit's dermatology related forums, and intervenes with text that is meant to convey the right information. However, their focus is on identifying the misinformation and not necessarily on the text-generation as that is made up of templated responses with general facts about essential oils.

More along the lines of generating humour, Chen and Eger (2022) use Seq-to-Seq transformers fine-tuned on a newly annotated dataset to generate funny titles for papers given their abstract, and compare performance against both humans and ChatGPT.

Another effort, which has more focus on text generation is by Çetinkaya et al. (2020). They build a model using state-of-the-art architectures, to both classify a tweet as well as generate a reply that either agrees or disagrees with the original tweet. The model is trained and tested on the gun control debate in the United States, and the bots are trained to first classify if a tweet is for or against gun control. Following the classification, a bot from either side generates a tweet reply that either agrees with its side, or provides a meaningful response to a tweet from the opposing side. They also propose a new salience metric, to evaluate the relevance of the generated tweet to the original.

There are also classification tasks that have been performed on Reddit data, one of which is described by Tang et al. (2022) where they attempt to classify a comment into a category of humor. This work involves finetuning a transformer model on Reddit data and is an inspiration for some methods used here.

Finally, there is a thesis by Willemsen (2017) in which the author trains a sequence-to-sequence model to reply to Reddit comments. They use the same dataset (Baumgartner et al., 2020) intended to be used in this project, and employ different kinds of models. They evaluate models with character-level representations, word-level representations, and hybrid-level representations to get the best of both worlds and present a comparative study. They also use a retrieval based model that uses a weighted score to pick the best reply from a collection of replies as a gold standard to compare against. They also use multiple evaluation techniques like computing scores for BLEU, ROUGE, and CIDEr, while also critiquing overlap based evaluation criteria as being ineffective for these kinds of generative tasks as the potential responses span a large space.

All of these ideas tie into the goals of this project, and using these works as references will aid in its progress.

4 Methodology

The data was cleaned to have the post title concatenated with the post body, and then paired with popular (> 100 upvotes) top level comments for that post.

I use a sequence-to-sequence transformer model similar to that described in Willemsen (2017). The rationale behind this methodology is that in order to fully use the information in the post to generate a witty response, this task can be viewed as a translation task. Variants of the T5 (Text-To-Text Transfer Transformer) HuggingFace models were used.

4.1 Baseline: Unigram Language Model

As a baseline model to compare performance against, I build a unigram model on the comment corpus, and have this model generate texts of arbitrary lengths without context of the Reddit post.

4.2 Baseline: Pre-determined comments

I use another "model" which randomly selects a comment from a predefined set of funny comments and posts that as a response to a post. This was mainly built to compare how many upvotes a generic comment would receive against a language model that was generating free text. Examples of sample comments are "F", "That's what she said" and "That's enough internet for today" among others which are popular quips on Reddit.

4.3 Prompt Based Model

Using the google flan-t5 model as a starting point, a few prompts were written to coax the model into providing a witty response to the post. An example prompt was as follows:

```
If you were a witty person,
how would you reply to: {}
```

The prompt to select is randomly chosen at run time.

4.4 Finetuned T5 Model

The t5-small model was finetuned with the Reddit dataset described above and then evaluated in the hopes of achieving better performance than the zero-shot prompt engineered model.

5 Evaluation and Results

Since this task is primarily based on text-generation, none of the classic evaluation metrics like accuracy or F1-scores are applicable. However, there are alternative evaluation metrics which are applicable for this project, two of which are detailed here.

5.1 Perplexity

The first is the perplexity score of the model for a test set. The idea of perplexity is that a "good" model should have a relatively high probability of generating sentences that generally fit the bill of "witty" and "humorous" and lower probabilities to sentences that do not.

I use 336 Reddit comments unseen by the model as a test set and compute the mean perplexity score across the test set, for each model. These mean perplexity scores are presented in Table 1.

5.2 Upvotes

Once the model is trained, and the bot is deployed, a live feedback loop is established where the users of Reddit are potentially upvoting or downvoting the comments made by the bot. An average score of these upvotes can potentially be an interesting metric to gauge the performance of the model.

Model	Mean Perplexity
Unigram LM	1892
Prompt-Based flan-t5	74
Fine-tuned t5-small	247

Table 1: The mean perplexity score was the best for the Prompt-Based flan-t5 model, while the finetuned model performed worse. Both models performed significantly better than the unigram LM, which was expected.

However, getting engagement from Reddit audiences proves to be a challenge. Most comments made by the bot are at 1 upvote (the default value) as the comment does not have enough visibility. However, one comment from the baseline model with predefined comments garned a large number of downvotes.

```
This is the good luck bot.
1 upvote = 1 good luck
```

A user responded with:

```
Negative good luck lessgoo
```

This metric was not ideal to judge performance due to the high uncertainty on whether a post would get visibility or not.

6 Discussion

While the perplexity scores provide a quantitative way of putting models into context, looking at comments generated by these models is the qualitatively more useful.

Neither model has generated a truly "funny" comment, that I would expect to get awards on Reddit. However, I present a couple of notable ones here.

```
I'm a bitty fucking fan of you guys
```

as a response to a r/ShowerThoughts post saying "Every comment section is a live chat". This mildly makes sense because the comment section is usually filled with a large number of people, and is known for the camaraderie they share.

For another post, that said:

```
People who refuse to look at arguments
objectively are usually objectively wrong
```

the prompt based model responded with:

```
Those who do not look at
arguments are usually wrong.
```

This prompt leveraged the summarization capabilities of the model and was shown the post with the "Summarize: " prefix.

Overall, it seemed like both models were largely indistinguishable from each other in terms of the actual comments they generated, while their perplexity scores had a significant difference.

In comparison to the unigram baseline, both models are significantly better as they generate cohesive English sentences which was something the unigram language model was unable to do (primarily due to being trained from scratch on a relatively small dataset)

The reasons for the transformer models not performing exceptionally well are still largely unknown. One potential reason for the fine-tuned T5's failure could be that the fine-tuning was not done on enough data. Also, the pretrained task I was attempting to leverage was the question-answering / reading-comprehension task where the input is provided in "question: " with "context: " format. This task is not ideal, but it was the most relevant on the tasks the model had been pretrained for. The question was written as "What is a witty response to this?"

This might actually have made the model worse, as the baseline mean perplexity (without finetuning) was around 100, which is lower than the score of 247 observed with the tuned model.

With this in mind, it would be reasonable to state the perplexity scores between 70 and 100 are near the baseline for Reddit comment generation (out of the box) and a finetuning attempt would need to go at least lower than those numbers in order to be considered successful.

7 Conclusion

This paper presents a Reddit bot that can comment on posts using a sequence-to-sequence transformer model. The two variants (with and without finetuning) were tried out and both perform largely similarly (and poorly) with respect to qualitative human evaluation.

Future directions for this project include finetuning with larger datasets, or even training a transformer from scratch on a large Reddit corpus so that it models Reddit language better.

All code for this project can be found on this github repository: https://github.com/chittaranjan19/witty-reddit-bot

8 Other things I tried

While I was able to successfully implement all the models I had originally envisioned, satisfactory

performance was something that was lacking from all models.

I tried playing around with the hyperparameters, running for larger number of epochs, using different templates for the prompt, but none of these had a large effect on the model responses.

9 What I would have done differently

I would have put more thought into the upvote evaluation criteria since I did not realize that user engagement would be a bottleneck. Upvotes are a great metric that encapsulate both qualitative and quantitative feedback, and it would have been good to have a framework that could efficiently enable collection of this data.

References

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *International Conference on Web and Social Media*.

Yusuf Mücahit Çetinkaya, İsmail Hakkı Toroslu, and Hasan Davulcu. 2020. Developing a twitter bot that can join a discussion using state-of-the-art architectures. *Social network analysis and mining*, 10:1–21.

Yanran Chen and Steffen Eger. 2022. Transformers go for the lols: Generating (humourous) titles from scientific abstracts end-to-end.

Monique A Sager, Aditya M Kashyap, Mila Tamminga, Sadhana Ravoori, Christopher Callison-Burch, and Jules B Lipoff. 2021. Identifying and responding to health misinformation on reddit dermatology forums with artificially intelligent bots using natural language processing: Design and evaluation study. *JMIR Dermatology*, 4(2):e20975.

Leonard Tang, Alexander Cai, Steve Li, and Jason Wang. 2022. The naughtyformer: A transformer understands offensive humor.

Bram Willemsen. 2017. *I Am A Sequence-to-Sequence Model trained on Reddit Data, Ask Me Anything! Generating Replies to Reddit Comments with Attentive Encoder-Decoder Networks*. Ph.D. thesis, Tilburg University.