Neural Text Generation Conditioned on Title-Verdict Pair from the Subreddit r/AITA

Eshwar P Sivaramakrishnan
esivaram@usc.edu

Krithika Karthikeyan
kkarthik@usc.edu

Supriya Sundar Raj
ssundarr@usc.edu

Swarali Atul Joshi
swaralia@usc.edu

Utkarsh Baranwal
baranwal@usc.edu

1 Introduction

In this project, we attempt to improve conditioned neural story generation. Specifically, given a title and a verdict, the task of our neural model is to generate a short, coherent first-person story that is not only consistent with the title but also adheres to the verdict. The verdict label chosen can either be one that sheds a positive or negative light on the narrator.

1.1 Motivation

Automatic storytelling is a particular area of NLP that is being researched extensively. Current state-of-the-art story generators are limited in the sense that they allow little to no input modeling from the user (Alabdulkarim et al., 2021). In models that do allow users to have a say, it is largely in the form of an outline or a prompt for the story. Giving users the flexibility to influence the overall direction or sentiment of a story is important since it could lead to better diversity in the generated story.

To make our neural models understand verdicts, we make use of data from Reddit. Reddit is a collection of communities (a.k.a subreddits) that are devoted to specific topics.The subreddit r/AITA(*Am I the A**hole?*) focuses on a peculiar story-telling archetype: one that begs for judgment by the court of public opinion. On r/AITA, users express morally gray personal situations from their lives as short narratives, while other users pass a verdict (Kassandra, 2019) on whether or not the narrator is at fault in the described situation.

1.2 Novel Contributions

An obvious application of the Reddit dataset would be to predict the verdict, given the narrative. Such a classification task is trite and overdone. Instead, we propose a neural text generation system that generates short compelling narratives conditioned on a title,verdict pair. A verdict provides a diversifying ‘seed’ and supplements text generation with additional context and direction. Depending on the verdict, the generated story can be drastically different, given the same title. For instance, a post that is titled “AITA for leaving my partner?”, can lead to the generation of completely different narratives depending on whether the verdict is YTA[A1] or NTA[A1]. Additionally, we use an anchor-based approach for gaining control and ensuring event coherence in the generated text.

2 Related Work

In this section, we review existing approaches for automatic text generation conditioned on inputs. (Alabdulkarim et al., 2021) discusses a number of approaches to text generation such as reinforcement learning, plot machines, and generation by interpolation. Reinforcement learning uses reward functions to control the ending of a story. The plot machine approach utilizes discourse structures to keep track of the generated text. Generation by interpolation generates text by conditioning on the first and last line.

A key challenge is controlling the flow of the generated text such that it adheres to the given inputs. (Fan et al., 2018) tackles this challenge by training a hierarchical model that generates a story in two phases- first create writing prompts and then generate a story based on these prompts. (Yao et al., 2018) also leverages a hierarchical approach by first generating a storyline and then using the storyline to generate a full text.

Another challenge is evaluating the performance of the text generation model, as there is no single ground truth for such an unconstrained task. (Fan et al., 2018) proposes prompt ranking accuracy, a method that measures the extent to which a generated text depends on its inputs.

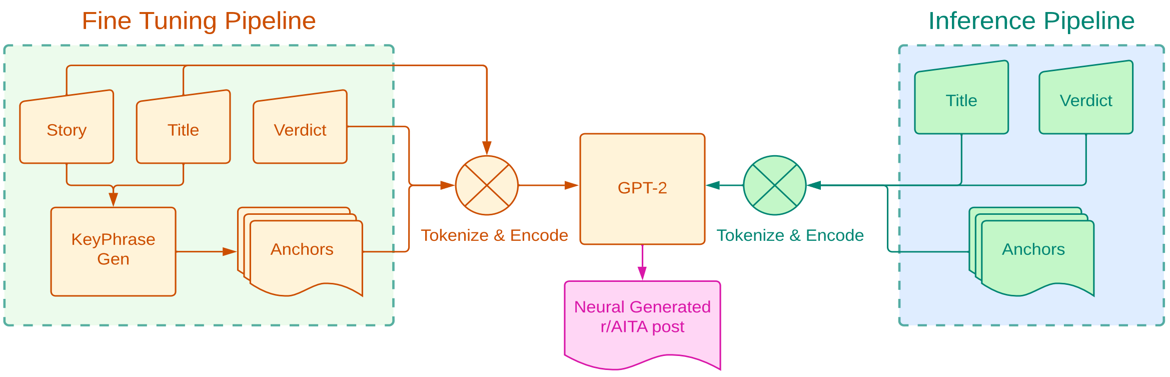

Figure 1: Training Pipeline

3 Method

3.1 Dataset

To power this corpus-driven application, we use Reddit API(Reddit, 2019) to extract 121,634 posts from r/AITA between January 1, 2014 and January 1, 2022.

In accordance with our data exploration, we perform the following preprocessing steps:

1. Retain only those posts which have more than 5 upvotes to make sure that posts and the corresponding verdicts are meaningful.
2. Retain only those posts with NTA and YTA verdicts.
3. Downsampling to balance the dataset between the classes YTA and NTA (originally, YTA-26%, NTA-74%).
4. Retain posts with over 105 and under 596 tokens (5th - 95th percentile of word token distribution in the dataset).

The resulting dataset consisting of 38,714 posts is divided into two parts:

- **Training Data** - 37,751 posts
- **Inference Data** - 963 posts

Further, we use two methods to extract keyphrases for each post:

3.1.1 RAKE

RAKE (Rose et al., 2010) is an unsupervised, stochastic, domain and language independent keyphrase extraction method. It is based on the observation that standard keywords rarely contain standard punctuation and stop words.

3.1.2 KeyBERT

KeyBERT (Grootendorst, 2019) is a BERT based key phrase extraction method that tries to extract keyphrases which have a similar embedding to that of the complete text or document. In our project, we use keyBERT with n-gram range of 1 to 3, Maximal Marginal Relevance set to true and diversity set to 1.

3.2 Modeling Approaches

The key challenge to generate a compelling story is taking control of generation away from probabilistic language modeling and injecting a global context, so that the generated story pertains to the title and the verdict. We explore two main approaches and an auxiliary approach with a bigger pre-trained model. We outline all three of them in the following sections.

3.2.1 Naive Fine-Tuning

We fine-tune the pre-trained model – GPT-2 for the task of conditioned generation. GPT-2 is generally fine-tuned through preparing the prompt to include all the necessary conditioning inputs. This naive fine-tuning pipeline can be imagined similar to the one in Figure 1, without the anchors in the input. In essence, the triples *{Title, Story, Verdict}* are provided as input in the form of a single concatenated prompt, using special tokens <*endoftitle*>, <*endofverdict*> and <*endoftext*> as separators.

3.3 Anchor Based Approach

The performance of the naive fine-tuning approach falls short of expectations, mainly due to the problem of inter-story repetition (refer to [results] section). To keep event-coherence intact and to control the flow of the story, we introduce anchors as additional training inputs. For this fine-tuning task, the prompt is a concatenation of the inputs Title, Story, Verdict, Anchors for each sample. Additional special token <*eok*> is used as a separator for the anchors. Two types of anchors are explored in the fine-tuning task, namely, RAKE anchors and KeyBERT n-grams (as discussed in the previous section). The fine-tuning pipeline can be found in Figure 1.

3.4 Fine-Tuning GPT-Neo

GPT-Neo is an open-source and parallel implementation of GPT-3 that has been implemented using the mesh-tensorflow library (Tensorflow, 2018). The motivation behind fine-tuning and testing the GPT-Neo model is to determine if the quality of automatic story generation is significantly dependent on the size of the model.

The base model is fine-tuned to generate a story, given the <*title, verdict*> pair. GPT-Neo works remarkably well in zero and few-shot scenarios and does not require complex prompt engineering with token separators. Our prompts for GPT-Neo are just the title, verdict and story on 3 separate lines.

4 Experimental Setup

4.1 Baseline Methods

Our baseline model is GPT-2 fine-tuned on the inputs {title, verdict, story}.

4.2 Training Setup

We fine-tune GPT-2 (with 117 million parameters) and generate stories using top-k (with k=50) and top-p (with p=0.95) sampling (Holtzman et al., 2019). We find this sampling method more effective at producing diverse phrases than beam-search. GPT-Neo (with 1.5 billion parameters) is fine-tuned on TPUEstimator with the GELU (Gaussian Error Linear Unit) activation function. The training hyperparameters are listed in Table 1.

Hyperparameters	GPT-2	GPT-Neo
Learning Rate	5.0E-04	2.0E-04
Sequence Length	800	2048
Device	GPU	TPU
Epochs	4	6

Table 1: Hyperparameters

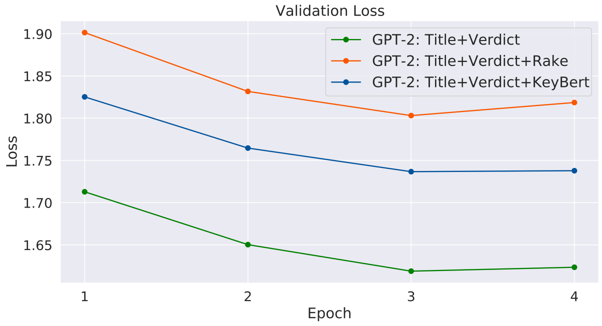

Figure 2: Learning Curves

The loss convergence graph for the trained models are presented in Figure 2.

4.3 Evaluation Protocols

Since the models perform open-ended story generation when given a unseen title and verdict pair, automatic evaluation metrics such as perplexity and BLEU may not be sufficient. We therefore include prompt ranking accuracy as an additional metric.

Perplexity: The confidence of the predictions made by the model is computed using perplexity. The perplexity of a sequence of tokens is calculated with the following equation:

$$Perplexity(X) = exp(\frac{-1}{t}\sum_{1}^{t} logP_{\theta}(x_i|x_{<i})) \tag{1}$$

Where X is the sequence, t represents the time step and x_i represents a token at time i. The overall perplexity is calculated by averaging over the whole document.

BLEU Score: It calculates the amount of n-gram overlap between the reference and the model's generated text.

Prompt Ranking Accuracy: Prompt ranking accuracy (Fan et al., 2018) quantifies the dependence of generated text on the given prompt. It does so by calculating the conditional probability of a story having been generated given the original prompt versus nine other fake prompts.

5 Results and Discussion

5.1 Results

We analyze the effects of our modeling improvements through the introduction of anchors on the

Model	Inputs	Perplexity	BLEU	Prompt Ranking
GPT2 Fine-tuned	Title + Verdict	17.67	0.304	0.09
GPT2 Fine-tuned + Rake anchors	Title + Verdict + Rake keywords	11.45	0.365	0.10
GPT2 Fine-tuned + KeyBERT	Title + Verdict + KeyBERT N-grams	12.17	**0.380**	**0.12**
GPT - Neo	Title + Verdict	**5.40**	0.194	-

Table 2: Results

inference dataset.

5.1.1 Fine-Tuning on Title-Verdict Pairs

The baseline results are expectedly inferior to other approaches, which can be seen in Table 2. The naive, fine-tuned model suffers from a short context horizon problem, a previously documented issue with pre-trained language models (See et al., 2019). The short context horizon problem, also known as the inter-story repetition problem, refers to the phenomenon of language models repeating specific n-grams over and over. The problem is generally attributed to a lack of probability diversity while sampling for the next word to be generated.

5.1.2 Fine Tuning on Rake Anchors

Keywords extracted with RAKE act as good anchors to maintain context. This is supported by the improved Perplexity, BLEU score and Prompt Ranking Accuracy.

5.1.3 Fine Tuning on KeyBERT Anchors

The n-grams generated by KeyBERT are contextually important keyphrases. These n-grams lead to comparable performance to the RAKE version in terms of BLEU and Prompt Ranking Accuracy, although it leads to a slightly higher perplexity score.

5.2 Discussion

5.2.1 Generation Quality

Our proposed anchor based model exhibits the capability of producing unique stories that follow a similar language to the posts made on the particular subreddit, characterized by a personal and accusative tone. Many of the posts made on the subreddit r/AITA also have this accusative tone and are usually an attempt to absolve oneself from blame. A reduction in perplexity score from the baseline is indicative of this. However, the model has its drawbacks which we elaborate below.

5.2.2 Inter-Story Repetition

The anchor-based models demonstrate that additional context grounding through the use of keyphrases alleviates some inter-story repetition. Despite this, as the maximum length of a story grows, the amount of inter-story repetition grows proportionally.

5.2.3 Performance Compared to Bigger Models

Despite fine-tuning and additional context from anchors, GPT-2 based models do not outperform bigger pre-trained models (such as GPT-3, GPT-Neo with billions of parameters). The bigger pre-trained models are generally able to write stories that exhibit more likeness to reddit language, indicated by a significantly low perplexity. They also write stories that are better in terms of diversity, observable from a lower BLEU score.

6 Conclusion & Future Work

In this project, we explore conditioned automatic story generation in the context of reddit posts. We propose and explore two anchor-based approaches in addition to simple fine-tuning of pre-trained models to achieve better event-coherence in the generated stories.

Our current fine-tuning approach lacks commonsense awareness. In the future, we plan to explore better common sense grounding techniques (Mao et al., 2019). It can also be noted that a generative task such as ours is faced with a lack of sophisticated automatic evaluation metrics. As an extension, we plan to include human evaluation metrics such as the triplet pairing task (Fan et al., 2018). Finally, we hope to try more sophisticated key-phrase extraction and topic modeling approaches to improve the quality and diversity of the anchors.

Team responsibilities

- Exploratory Data Analysis & Preprocessing : Utkarsh, Swarali
- GPT-2 baseline + anchors: Eshwar, Krithika
- GPT-Neo: Supriya
- Evaluation Metrics: Krithika, Eshwar
- Final Report : Everyone
- Poster Presentation: Everyone

References

Amal Alabdulkarim, Siyan Li, and Xiangyu Peng. 2021. Automatic story generation: Challenges and attempts.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation.

Maarten Grootendorst. 2019. keybert.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration.

Lopez Kassandra. 2019. A flourishing community of potential assholes.

Huanru Henry Mao, Bodhisattwa Prasad Majumder, Julian McAuley, and Garrison Cottrell. 2019. Improving neural story generation by targeted common sense grounding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5988–5993, Hong Kong, China. Association for Computational Linguistics.

Reddit. 2019. Pushshift reddit api documentation.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. *Automatic Keyword Extraction from Individual Documents*, pages 1 – 20.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do massively pretrained language models make better storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, Hong Kong, China. Association for Computational Linguistics.

Tensorflow. 2018. Mesh tensorflow - model parallelism made easier.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2018. Plan-and-write: Towards better automatic storytelling.

A Appendix

A.1 Acronyms

1. YTA - You're The A**hole
2. NTA - Not The A**hole

A.2 Sample generated story

Title: WIBTA for not showing my birth certificate when asked?

Verdict: YTA

Story: As a Japanese living in the USA, my next door neighbors are constantly asking me to go back to the Japan over the summer. They also like to make fun of me for not having a Japanese passport. They brag to my family about how they learned English as a child but I our native language is still English. Even with my English I can't communicate with them very well, and my family often me they are too busy7me to listen to my side of the story.

One day I was on the train home from work, and the neighbors were talking with my neighbor and was pointing her phone at me7and talking about how jealous she is that I'm not a Japanese. 8She even screams out loud about how she's always touchy with me and how I should come and visit her, even though she knows I'm not a Japanese.

7It's almost like they hate me because I don't have a Japanese passport.

I feel like their behavior is ridiculous, but at the same time, it is disrespectful to ask me to go back to Japanese Main society when my employer has me meet with them.

WIBTA if I told them I have nothing to do with Japan and even though I came here for tax reasons, I won't be showing them my birth certificate while they make unnecessary comments about it?