# Using GPT-2 and Deep R.L. from Human Feedback For Academic Topic Recommendation

BRENNAN RICHARDS, University of Rhode Island, USA

Recommender systems are useful for extracting valuable information from pools of large volumes of data. Considerable work has been done on making personalized recommendations based on user preferences and, more specifically, on allowing a user to define the reward function through interactions with the machine learning algorithm being used. Language models such as OpenAI's GPT series have also been rapidly advancing to gain general capability on a wide range of language tasks. This work combines these lines of research, using a GPT-2 model, which has been fine-tuned on the task of generating a new academic topic given a sequence of topics, as the policy network within a proximal policy optimization setup. A user interface is employed to return new topics to the user and allow the user to rate the topic in order to encourage or discourage similar return values in the future.

Contents



Author's address: Brennan Richards, brichards88@uri.edu, University of Rhode Island, 45 Upper College Rd, Kingston, Rhode Island, USA.

## 1 INTRODUCTION AND MOTIVATION

The goals of this project are threefold:

(1) Create a product which takes, as input, topics that a researcher is already interested in and returns a novel topic-of-interest each time it is called.
(2) Return a handful of research documents related to that topic along with some useful meta-information about those documents.
(3) Equip the system to learn, over a series of time steps, to recommend increasingly better topics according to the user's inputs and feedback.

## 2 RELATED WORK

GPT-2 [4], a natural language generation model from OpenAI, is used to take the researcher's interests as input and generate a title of a new research topic via natural language. This model uses a transformer architecture, and takes a sentence embedding as input, repeatedly predicting and appending the next word in the sequence to generate sequences of text. The model is trained via self-supervised learning on raw text and can be fine-tuned to specific tasks beyond the generally capable base model.

OpenAI's work on deep reinforcement learning from human feedback was also inspirational to this project, and a similar approach is followed. In their paper *Deep Reinforcement Learning from Human Preferences* [2], Christiano et al demonstrated that it is possible to teach a randomly initialized agent using proximal policy optimization to complete a back-flip strictly from human feedback. The agent generates a number of movement sequences and periodically asks the human judge to "score" sequences by selecting which of two generated sequences of movements is closer to the desired behavior.

In their work, the researchers at OpenAI train a second neural network in a supervised fashion to approximate the implicit reward function by learning the mapping between a generated sequence and the user's score. This important step speeds up the time to train the agent as the training loop can happen offline in the background on high powered GPUs by calling the reward network to predict how "close" a generated sequence of movements are to the desired behavior, and has not yet been leveraged as a part of this topic recommendation project.

## 3 METHODOLOGY

### 3.1 Data

*3.1.1 CiteULike.* Wang et al [6] contributed a key data set which contains a number of users and articles, as well as the articles that each user has saved to their profile and the "tags" they have attached to each article. For this project, the data set was engineered to have pairs of $x \in (v, \theta)$ where $v$ is a user identifier and $\theta$ is a list of tags that the user has used on articles saved to their profile. The list of tags are treated as a list of topics that the user is interested (indeed they often correspond to topic names i.e. "machine-learning", "deep_learning") and through the formatted data, a proxy is gained as to what topics each user is interested in. In this format, approximately 5,000 training points are available.

Because the order of the sequence does not matter, the tag lists are augmented to create a random number (between one and six) of random permutations of each tag list, which are appended to the data set which was then randomly shuffled. After de-duplicating, the resulting 15,000 or so rows are aggregated together into one large corpus of unsupervised text, and chunked into batches in order to fine tune the GPT-2 model to predict a new topic.

*3.1.2 OpenAlex.* OpenAlex is an open source database of research concepts, articles, institutions and authors built by OurResearch, a nonprofit open source research fund [3]. When the GPT-2 model generates text that it believes to be a reasonable topic based on the inputs, it send this text to the OpenAlex API via HTTP to check for similar concepts. If the concept exists, a number of research articles from this topic/concept are also retrieved in order to return to the user.

*3.1.3 Human user's reward signal.* The final source of data for this project is the reward signal gathered from the user. At each time step, $t_n$, the model interacts with the user to ask for a reward, $r_{n-1}$, which is a floating point number between -1 and 1 and corresponds to the user's score for the recommended topic output of the model at the previous time step $t_{n-1}$. In cases where the output of the model is unintelligible or otherwise cannot be matched to a concept in the OpenAlex database, the reward is set to -1, to encourage coherent outputs.

### 3.2 Algorithms

*3.2.1 Non-standard data cleaning and filtering.* Along with standard data cleansing techniques for natural language data sets, such as removing stop words, the tags were cleaned to remove non-alphanumeric characters such as the *dash* and *underscore* characters that are found in a majority of the raw tags. A python package called *wordninja* was also used to split text without spaces into sequences of words, for example an input of 'cleaningandfiltering' becomes 'cleaning and filtering'. This method is imperfect, and sometimes introduces short and meaningless word parts into to the data set, which were removed in downstream iterations of the data set when discovered.

The tags in the engineered data set were filtered by using a BERT-based sentence embedding model from the Hugging Face Transformers library [7] to compute a Euclidean distance, which doubles as a similarity measure, between each topic and a handpicked list of science, technology, engineering and mathematics topics (i.e. "machine learning", "mathematics", "science") using the following algorithm. For each tag in each user-tag list:

(1) Embed the tag using the BERT-based sentence embedding transformer from Hugging Face.
(2) Compare the embedded tag to each handpicked topic in the list by computing the Euclidean distance between the two vectors.
(3) If the distance between the current tag and any of the handpicked topics is closer than the threshold, keep the tag in the data set. Otherwise, discard it.

The distance threshold was chosen such that approximately 75% of the tags in the data set were filtered out. The result is that the system is specialized on recommending new S.T.E.M. topics from S.T.E.M. topic inputs. Future iterations could use a looser similarity threshold in order to serve as a more generally capable recommender across a wide range of academic topics.

*3.2.2 GPT-2 from Hugging Face Transformers.* The GPT-2 model used comes from Hugging Face Transformers [7], a library of pre-trained neural networks that use the transformer architecture. The model was instantiated in Python as a Keras model and then fine-tuned on the engineered CiteULike data set such that given a list of academic topics of interest, the model returns a new topic.

*3.2.3 Proximal Policy Optimization.* The GPT-2 model was then used as the *policy* in a proximal policy optimization (PPO) algorithm. PPO is a deep reinforcement learning method which iteratively trains a policy network to take actions which will maximize a reward returned by the environment in question [5]. Instead of being trained on a *loss* function, the PPO algorithm can be thought of as using stochastic gradient ascent to learn the parameters of the policy network which lead the agent to take actions that maximize reward. Used in this way, PPO becomes a second fine-tuning step,

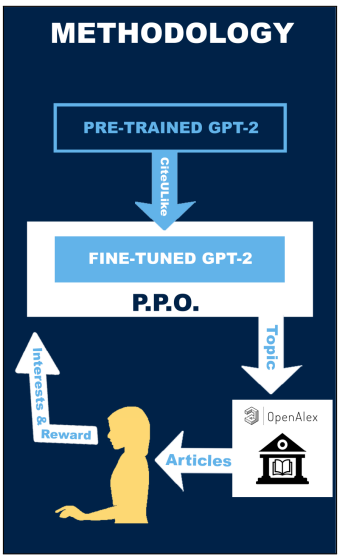


Fig. 1. The methodology behind the system. (1) A pre-trained GPT-2 model is fine tuned on the engineered CiteULike data set. (2) The fine-tuned GPT-2 model becomes the policy of a PPO algorithm. (3) The recommended topic output at each time step is fed into the OpenAlex API to retrieve a matching academic concept and articles related to that concept. (4) The user interprets and score the output, and the reward is fed into the PPO algorithm which optimizes GPT-2 to output better topics.

fine tuning the GPT-2 policy from one that recommends academic topics to one that recommends academic topics that please a specific user.

As noted prior, the GPT-2 model was fine-tuned by loading the transformer into a Keras model (a fantastic feature of Hugging Face). The PPO algorithm use was part a Python library titled *TRO* which was built for the express purpose of using Hugging Face transformers, though thus far only GPT-2 is stably implemented, within a PPO deep reinforcement learning algorithm. Because this library is written with Pytorch, the Keras GPT-2 model had to be converted into a Pytorch model, which is easy to do with the *Hugging Face Transformers* library when loading a model from a pre-trained checkpoint (just include a from_tf=True flag).

*3.2.4 User Interface.* Figure 1 gives a conceptual overview of the entire algorithmic framework. A user interface is built, using the *gradio* [1] Python package, to deliver topic recommendation and related content to the user and query the user for a reward. One time step of the user interface is shown in figure 2.

## 4 RESULTS

### 4.1 Model Output Examples

The table below demonstrates the difference between outputs of a GPT-2 model fine-tuned on 20 human interactions and a model that has not yet experienced any human feedback. Each model was given a list of input topics and queried five separate times to generate continue on the sequence of topics. The output sequences are pruned to three topics each, but many ran longer.

Note that the model that is fine-tuned on user feedback is less likely to generate nonsensical text, although both models exhibit this problem at times. This may reflect either general variability in the GPT-2 model or, more likely, subtle errors in the data cleansing process.

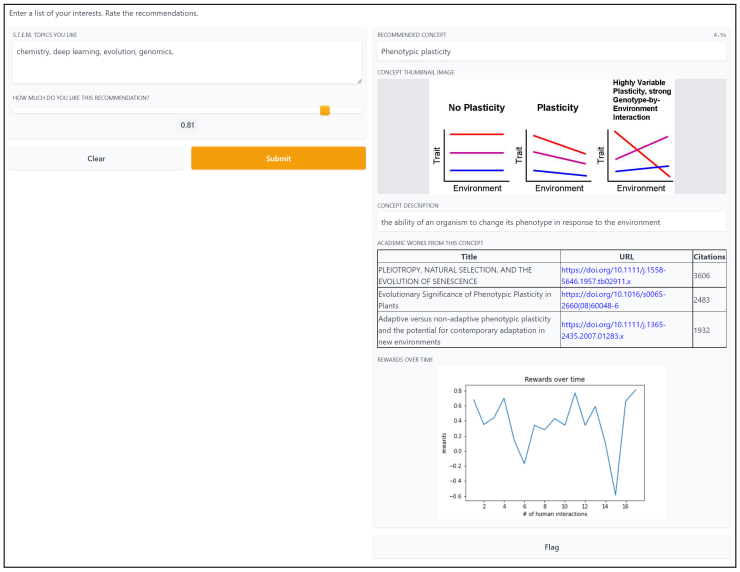


Fig. 2. Gradio interface for interacting with the final system. (Left) The user enters a comma-separated list of topics that they are interested in. The interface includes slider [-1, 1] for the user to indicate a score for the returned topic. (Right) The model generatively infers the next topic(s) in the sequence and queries the OpenAlex database for a concept, description, image and related articles. The UI also plots rewards over time for evaluation of the algorithm's improvement.

| GPT-2 Output Given: "chemistry, deep learning, evolution, genomics, statistics, graph theory," | | |
|---|---|---|
| Run | Fine-tuned on 20 user interactions | W/o user feedback |
| 1 | software exploration, educational experience, sense | user experience, information extraction systems, statistical concepts |
| 2 | recognition, technological determinism | t, v, user attention |
| 3 | concept identification, a algorithm, network neural | quantum information processing, parietal cortex, intelligent systems |
| 4 | robot scientistlecture, riemann, usabilitycom cogartical | e, fagins algorithm, s |
| 5 | pro, survey paper, criterion | probabilistic relational models, genome project, computer network |

### 4.2 Fine-tuned GPT-2 for STEM Topic Recommendation on Hugging Face

The GPT-2 model fine-tuned on the academic topic data set is available and open sourced for download using the Hugging Face Transformers library at: https://huggingface.co/brennan-richards/gpt2-finetuned-academic-topics. The model is available as either a Pytorch or Keras transformer.

### 4.3 Google Colab notebooks

The Google Colab notebooks through which this project was developed are available for viewing to anyone with a University of Rhode Island Gmail account.

- Data set engineering: https://bit.ly/3wgHWa3
- Fine-tuning GPT-2 on academic topics: https://bit.ly/38eKkpG

- Fine-tuning academic topics generator on human feedback: https://bit.ly/3P4zmnz

## 5 CONCLUSION AND FUTURE WORK

### 5.1 Future Work

The major next step down this line of work is to instantiate a second neural network and train it to infer the reward function from human feedback. The inputs to this network would be the sequence of topics that the user provided as input, and the GPT-2-generated continuation of this sequence. The outputs would be a floating point value representing the reward. With this network, the GPT-2 network used as the PPO policy can be fine-tuned on the inferred reward function in the background by querying this reward-learning neural network instead of the user. In this way, many hours of human interaction time can be offloaded onto numerous GPUs running in parallel to compute a better policy.

Due to the random tokens which sometimes appear in the outputs of the GPT-2 models, developing another improved methodology for data set cleaning seems to be a promising direction for improvement. One possible feature of this new cleaning method would be allowing a wider range of tags to pass the filter into the training data set. Another could be filtering out non-English words (usually a compute-intensive process).

Larger, newer models such as GPT-3 (only available through API) and GPT-J (open source) may also improve the performance of the system overall. Using the available models would take a new set of libraries, or some hacking of the source code of the tools currently in use.

Finally, the system as a whole could be improved by attaching it to a web application similar to the original CiteULike website, so that new user data can be repeatedly gathered, stored in a database, and added to the training data set. This data can then be used to create improved versions of the fine tuned topic recommender which can then replace the policy network in the PPO framework.

### 5.2 Conclusion

In this article, a new approach to AI-assisted research discovery was proposed. GPT-2 was leverage to generate research topics based on a user's interest after being fine-tuned in an unsupervised fashion on a data set of user-tag pairs. This model becomes the policy in a PPO algorithm, which works to fine-tune the GPT-2 policy once again, this time on the reward signal from the user. The result is a generative model that takes a user's interests as input and outputs promising new lines of research, which have already taken up a number of spots in the author's bookmarks bar.

## REFERENCES

[1] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. 2019. Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. *arXiv preprint arXiv:1906.02569* (2019).

[2] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf

[3] Jason Priem, Heather Piwowar, and Richard Orr. 2022. OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. https://doi.org/10.48550/ARXIV.2205.01833

[4] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).

[5] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. https://doi.org/10.48550/ARXIV.1707.06347

[6] Hao Wang, Binyi Chen, and Wu-Jun Li. 2013. Collaborative Topic Regression with Social Regularization for Tag Recommendation. In *IJCAI*.

[7] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6