Topic Controlled Language Models using Contrastive Experts

Jordan Voas
University of Texas at Austin
jvoas@utexas.edu

Alex Chandler
University of Texas at Austin
alex.chandler@utexas.edu

Abstract

We explore the use of contrasting expert language models for the use of topical control on language generation. As a result, we examine methods to extract linguistically significant meaning from self-organizing structures trained on a large corpus of Wikipedia-drawn text summaries and their corresponding categories. Our findings show extracting such meaning is difficult and needs further work, with the contrastive experts expressing opposing probability adjustments that simply cancel each other out. Further, when using only our positive expert as opposed to the contrastive experts, we observe metric improvements for our in-domain data but a failure to match pre-prompting baselines on out-of-domain data. Our work is a novel exploration of self-organizing structures for this purpose and presents a starting point for such methods going forward.

1 Introduction

Large language models have grown increasingly powerful and expressive with recent advances. Despite these improvements, many models do not natively incorporate controllable mechanics to steer generation other than providing a leading prompt that can produce unnatural or over specified results (Zou et al., 2021).

Furthermore, many language models struggle with long-form generation and fail to return to topicality if a portion of text produced does not contain the tokens corresponding to the desired topic. This occurs because many architectures have a historical size limit on each generation inference step (i.e. GPT-2 (Budzianowski and Vulic, 2019) can only consider the 1024 most recent tokens).

Attempts at adding controllable mechanisms to large language models typically fall into one of two categories - those that require fine-tuning of the language model, and those that do not. While fine-tuning approaches appear to be simple on the surface, this requirement has several practical implications. As language models grow in complexity, fine-tuning necessitates massive amounts of memory, data, and time. Fine-tuned methods also fail to transfer to formulations outside of their training specification without further fine-tuning.

Alternatively, recent approaches adjust language latent token probabilities using the output of secondary models or conditions. These methods are easier to train, as smaller secondary models can often be used. This provides more flexibility and modularity by being capable of swapping out augmenting models, and has been shown to be competitive on absolute performance with fine-tuning approaches.

In this paper, inspired by the DExperts work using contrastive co-expert language models (Liu et al., 2021a), we attempt to develop an augmentative approach for the generation of topicality-controlled text. We aim for our approach to provide expansive and general topicality transfer capability while remaining capable of being applied to any shared vocabulary language models, even when the expert model is based on a simpler architecture. This approach differs from similar works in the goal of utilizing a single expert model for generalizable topic control. Many works in the area of language model control target only a single label from a binary set of opposing categories to control for (i.e. toxic/nontoxic or positive/negative sentiment) or can be applied to only a constrained set of topic labels.

Our targeted contributions in this project include the following.

- Collection of a large web-scraped topic-diverse dataset based on the corpus of Wikipedia summaries.
- A novel method for diverse topic spatial clustering with an incorporated mechanism for topical inversion.

- Development of a topic-conditioned expert language model which attempts to generalize to extracted embeddings from diverse category sets[1].
- A evaluation of the DExperts control approach when applied to the non-binary and unconstrained topicality problem formulations through the use of the previously developed models.

2 Related Work

Here we will briefly review some related work that is relevant to our own.

2.1 Controllability for Large Language Models

It remains challenging to control long-horizon text generation. Existing work in language control has largely focused on controlling the polarity of text sentiment or reducing bias language generation (Sudhakar et al., 2019), (Li et al., 2018). (Liu et al., 2021b) created a contrastive expert model that combines a larger pre-trained language model with a smaller "expert" and "anti-expert" language model. DExpert shifts the probabilities of the larger pre-trained expert model by encouraging text considered likely by the "expert" LM and discouraging text considered likely by the "anti-expert" LM. DExpert outperformed all existing controllable generation methods such as the PPLM generation control method (Dathathri et al., 2019).

2.2 Self Organizing Map Models

A Self Organizing Map (SOM) (Kohonen, 1990), or Kohenin's map, is a type of artificial neural network used for unsupervised learning. It can be used for dimensional reduction, visualization, and clustering of high-dimensional data. In a SOM, the network learns to group similar input data together in a grid of interconnected nodes. Each node represents a high-dimensional vector, and the grid encodes the relationships between the nodes in a lower-dimensional space. The SOM training algorithm is an iterative process, where the nodes are adjusted in response to new input data, learning the underlying structure of the data. The result is a map that can be used to visualize and analyze the relationships between the input data.

2.3 Adapters and Auxiliary Text Transformer Conditioning

While task specific discriminative fine-tuning on large pre-trained language models has shown to yield strong performance (Radford et al.), fine-tuning the latest large language models is computationally intensive and parameter inefficient. (Houlsby et al., 2019) introduced a new solution to transfer learning with the adapter module. Adapter modules are compact, task-specific neural network modules that can be added to a pre-trained language model. These modules allow the pre-trained model to fine-tune its weights and adapt to the specific characteristics of the new task, without the need for large amounts of additional training data. An adapter module consists of a mixture of linear layers, normalization layers, and non-linear layers[2].

3 Dataset

In order to train our various model architectures, a relational dataset of on-topic text to multi-category textual labels is required. Furthermore, in order to maximize topic transferability we desire a dataset containing diverse and unconstrained categories. However, we are unable to locate any large and diverse datasets meeting these specifications.

For this reason, we create a novel dataset called Wiki Text to Multi-Category (WTMC) that includes 100,000 textual summaries and their set of categories collected from Wikipedia. Our collection of text summaries was produced by performing a breadth-first search on categories and subcategories available through the Wikipedia Main Topic Classifications page[3]. Wikipedia Topic Classification pages include both subcategory hyperlinks and pages that correspond to the category of the current page.

We perform this breadth-first search using Selenium and the Media Wiki API[4]. We collect text to multi-category pairs until we have collected 100,000 pages of data. Performing a breadth-first-search rather than a depth-first-search enables our categories to be a representative sample from the diverse groups of categories available through Wikipedia.

[1] https://github.com/jvoas655/TCL-Experts-for-Control

[2] https://medium.com/dair-ai/adapters-a-compact-and-extensible-transfer-learning-m

[3] https://en.wikipedia.org/wiki/Category:Main_topic_classifications

[4] https://www.mediawiki.org/wiki/API:Main_page

We choose to include only the Wikipedia article summaries primarily due to the inherent nature of their topical focus. Manual evaluation of a subset of Wikipedia articles shows that the summary sections tend to discuss topics heavily influenced by the article's category labels. We ignore later sections of the text from each Wikipedia page, as later paragraphs tend to go off-topic, discussing and linking to adjacent but potentially topically confusing features.

4 Methods

We will go over our methodology for producing our category-embedding conditioned contrastive expert model, including details on both training and inference. Our methods are inherently not end-to-end and require the training/optimization of multiple models. While this is a disadvantage in terms of training complexity, it provides a significant advantage in our ability to optimize, evaluate, and iterate on each stage of our pipeline, and reduce the concurrent computational requirements. The full pipeline for our methods is shown in Fig 1.

4.1 Category Embeddings

The first stage of our pipeline involves converting the unique set of our dataset's textural category labels into a representative embedding format. Our initial method to do this involves utilizing a large pre-trained language encoder, E_c, such as RoBERTa (Liu et al., 2019) or T5 (Ni et al., 2022) to take each individual textual category to an embedding space. A series of hidden activations for these models can be selected for this embedding space, with varied options on hidden activation pooling or concatenation. This is represented by the following notation to produce a category embedding h_c from a textural category label X_c.

$$h_c = E_c(X_c) \tag{1}$$

No model training is required for this step, and category embeddings can be efficiently preprocessed in later steps. Confirmation of meaningful embeddings is required, as meaningful embeddings possess significant downstream advantages. As such, we trialed multiple encoder model/parameter combinations for this step of our pipeline.

4.1.1 Dimensional Reduction

Irrespective of the method used for obtaining category embedding representations, we explored whether the high dimensional space produced contained some level of noise that does not represent meaningful information. Additionally, we check if a dimensional reduction to the embeddings can both produce smaller models as well as remove unlearnable representations. We, therefore, explored techniques to reduce the dimensionality of our category embeddings through a trained autoencoder. Training on our later steps of the model pipeline revealed consistent performance improvements with the unreduced category embeddings, so the reduced versions were not used for the final evaluation.

4.2 Self Organizing Maps and Category Inversion

Our project's premise relies on the ability to identify contrastive categories. The primary category focuses on training an expert model to generate on-topic tokens, while the anti-category works to undo any stylistic effects the expert model has on language generation without affecting the on-topic nature of the generated language. To accomplish this, we had to develop a method for identifying categories that are inverse to each other, at least in the context of language generation. To find the inverse of categories, we utilized a Self Organizing Map (SOM) (Kohonen, 2004). A SOM operates by defining a spatially described set of nodes, each of which is connected with some other nodes to form a sparse undirected graph. The dimension of the graph's representation can be denoted as d_L, and each node will be associated with some initial value in a higher dimensional feature space of size d_H. The graph contains K nodes.

A method of Hebbian learning is utilized to fit the SOM to a dataset, repeatedly sampling the dataset which in our case was the set of category embeddings h_c. For each sample, the method finds which node in the SOM possesses an associated value nearest to the sample and then strengthens (moves the association) nearer. The association is also strengthened with the sample for nodes that are spatially near the winning node in the map's lower dimensional space, with some parameter options related to decay rates, distance scaling, and threshold.

After training suitably, the SOM will not only produce a quantization codebook that provides adaptive coverage of the sample set but will also spatially cluster nearby values in localized sections of the SOM's lower dimensional graph representa-

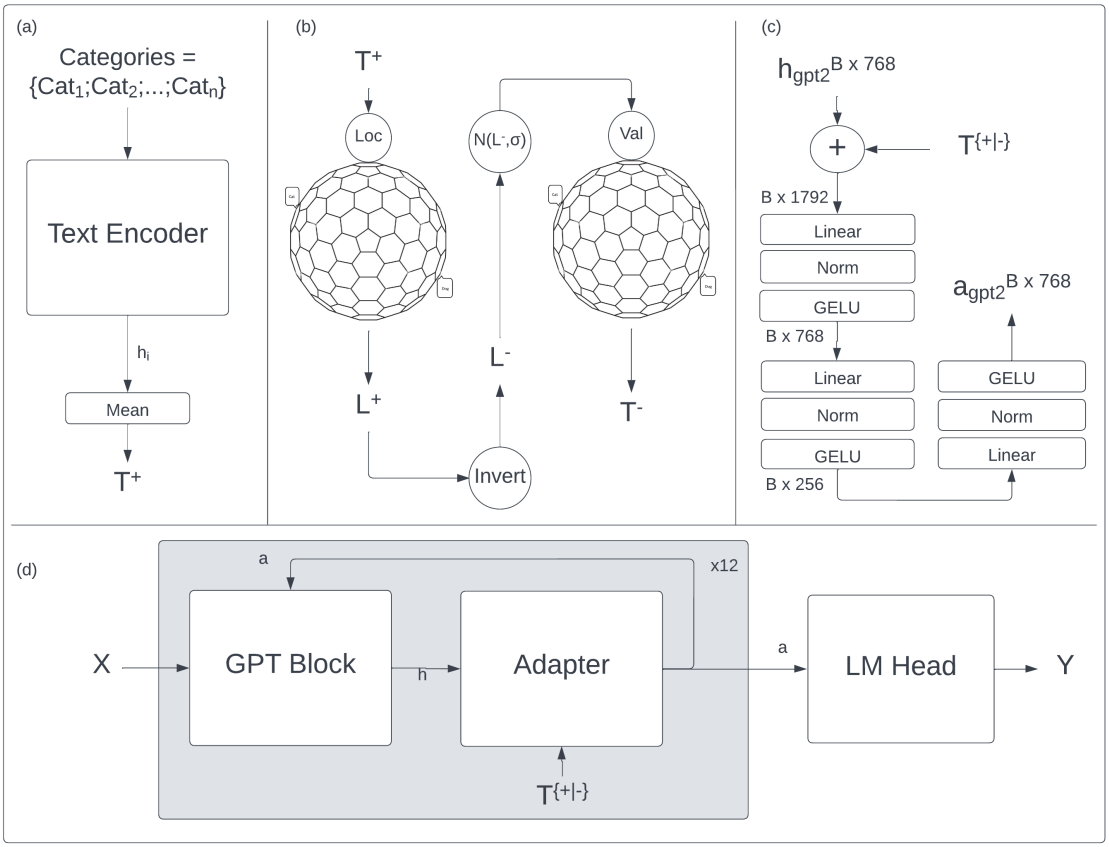

Figure 1: A illustration of our model architecture. a) The pre-trained category embedding extracting, taking textural appended categories to a fixed size embedding space obtained by performing a mean over the model's final hidden layers for all tokens. b) Our category embedding inversion methodology, taking the category embedding obtained from part a and localizing it on the spherical SOM. This localization is then inverted about the original and sampled around, from which the corresponding inverse node category embedding value can be obtained. c) Our category embedding conditioned adapter architecture. d) Our modifications to the pre-trained GPT-2 architecture to instill our category embedding conditioned adapters between each transformer block.

tion. In combination with the careful selection of our lower dimensional graph structure, this enables us to resolve the challenge of finding inverse category embedding values by selecting a structure that is pseudo-invertible. An example of such an occurrence is shown in figure 2. For any node $k \in K$ described by a topic encoding $h_T^{1\times K}$, we can find the node considered most contrastive by sampling one from the far side of the SOM.

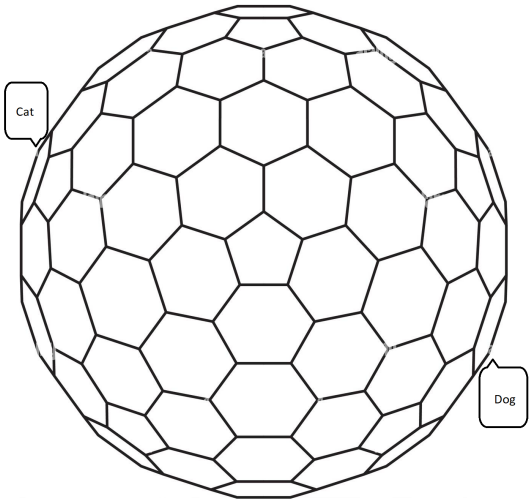

Figure 2: A example of a pseudo-invertible SOM structure in the form of a Goldberg Polyhedra

As our SOM is in the lower dimensional representation of a sphere-like structure, we reparameterize the distance between two nodes instead as the arc angle between them in radians, which provides a more linear learning structure.

4.3 Relational Quantization of Embeddings for Invertible Topic Encodings

We explored the use of topical encodings distinct from the source category embeddings extracted from the text encoders. To achieve this, we employed the trained Self Organizing Map (SOM) as a form of quantization codebook, C^K with K being the number of codebook entries allowed (Kohonen, 2004). To encode multiple textual categories, we used the nodes in the SOM as the codebook entries they correspond to and set a corresponding index from a binary vector.

This means for any category embedding, we were able to find which codebook entry it most resembles to produce a linear and sparse encoding of length k to represent it. Optionally, we trialed using variable values between zero and one, dependent on how well the embedding is described by the quantization. This was done multiple times for each category corresponding to a text summary to produce a fixed-length encoding of well-regularized and sparse values. We denote this topic encoding as $h_T^{1\times K}$. Generating inverse encodings was done by simply switching any set indices to those in the codebook which are spatially on the far side of the SOM, maintaining the total sum of activations between the pair of inverse encodings.

4.3.1 Trained Text-Based Topic Encoding Predictor

In order to utilize our SOM for the formation of topical encodings at inference times, where inputs may not have label categories, we desired a method to predict topic encodings from some initial small segment of text. This could be the initial segments of the generation prompt or an auxiliary prompt describing what the generation should be about. We thus attempted to train a topic encoding prediction model, M_T. This model was trained on the same corpus of Wikipedia summaries and categories used for training the SOM.

$$\hat{h}_T^{1\times K} = M_T(X_{topic}) \tag{2}$$

To train this model we utilized the following L2 loss.

$$\mathcal{L}_{prim} = -\|\hat{h}_T^{1\times K} - h_T^{1\times K}\| \tag{3}$$

We also incorporated a loss that penalized topic predictions similar to the corresponding anti-topic with the cosine similarity loss term $\mathcal{L}_{anti}$. This was done to further enforce the topical inversion property which was initialized in the unsupervised fitting of the SOM, and thus provide better generation contrast in our primary and anti-topic experts.

$$\mathcal{L}_{anti} = -S_c(\hat{h}_T^{1\times K}, anti(h_T^{1\times K})) \tag{4}$$

Our final loss for training our topic prediction model was $\mathcal{L}_{topic} = \mathcal{L}_{prim} + \mathcal{L}_{anti}$ and was built on the same large language encoder model used to obtain the category embeddings, but with appended task transfer layers.

However, we were unable to obtain any form of predictability for the topical encodings generated directly from the node activations of the SOM. Our model would inevitably converge to predicting the mean of all topic encodings to minimize loss. This led us to conclude that the topical encodings are either too spurious, with node activations having very little learnable significance with the summary text, or that the node activations have discarded too much information that was previously contained in the category embeddings.

4.4 Fine-Tuned Topic Conditioned Expert Model

After encountering the issues of directly predicting the topic encodings based on the text, we decided to forgo the prediction nature entirely, and instead utilize the category embeddings directly as our language model conditioning. These category embeddings can be obtained during inference directly from the deterministic and non-fine-tuned text encoder model, so we do not see the modification as a significant restriction. Based on this, we revised our inversion methodology but maintained our utilization of the SOM for the purpose. We instead obtain our inverted category embeddings by identifying the spatial coordinates of the node the primary embedding best maps to. This location can be routinely inverted about the origin to retrieve the nearest node's value to the inverted coordinate.

We then trained our single adaptive topic expert language model, $z_T = M_E(T, X)$, with T being a category embedding, X being a sequence of textual tokens, and z_T the predicted latent token probabilities. In the spirit of efficiency and following the work by DExperts, we utilized GPT2-small as the base of this model and appended the category embedding features through an adapter layer between each GPTBlock. In order to train the expert model, we provided the topic encoding for each sample drawn from our WTMC dataset and recurrently train it to predict the next token in the summary text based on the prior tokens.

4.4.1 Loss Function and Contrastive Reinforcement

Our loss function primarily attempts to promote the on-topic generation of the proper tokens through a cross-entropy loss comparing the category embedding conditioned generation results with the true labels. In order to promote contrastive behavior where possible, we also introduce a cross-entropy anti-loss by comparing the category embedding conditioned generation with the sampled tokens corresponding to an anti-category embedding. Our full loss is shown below, with the hyper-parameter λ controlling the strength of the contrastive focus.

$$\mathcal{L} = \lambda * \mathcal{L}^{+} - (1 - \lambda) * \mathcal{L}^{-} \tag{5}$$

$$\mathcal{L}^{+|-} = CrossEntropy(X, \hat{X}(T^{+|-})) \tag{6}$$

In addition, we implemented classifier-free guidance as well as probabilistic input token masking. For our classifier-free guidance, we randomly zeroed out the category embeddings for the generation in order to promote our model to generate fluent text independent of one of the category embeddings in the training set. Inversely, our input token masking was done by probabilistically masking out random individuals or chunks of tokens for the input sequence. This attempts to promote some level of reliance on category embeddings for generation. This is important for our training data set, as all samples are already on-topic, thereby running the risk of learning to determine the topicality of the generation more from the preceding prompt than the category embedding conditioning itself.

4.4.2 Anti-Category Embedding Corrective Mapping

Evaluation of the SOM node values and quantization error rates implied that, despite producing a well-separated spatial clustering on the training dataset, the SOM node values themselves are significantly out of distribution from the actual data on which they were trained. We theorize this is due to the SOM being forced towards the most restricted embedding mean which can separate the distribution. These values, if used directly, were observed to degrade the contrastive separation of the learned model even with very strong contrastive settings ($\lambda = 0.5$). To correct this distribution shift we utilize the training dataset to create a corrective mapping that takes each node in the SOM to the median training embedding which maps to it.

4.5 Contrastive Topic Conditioned Expert Inference

We followed the DExperts methodology at inference time with only slight modifications. Given a initial prompt X_{prompt} as well as a topic conditioning text prompt, X_{topic}, we can utilize our category embedding encoder model to find $\hat{h}_{T^+} = M_C(X_{topic})$. We then localized the anti-category embedding using our SOM through $\hat{h}_{T^-} = anti(\hat{h}_{T^+})$. At inference, we used no sampling on the inversion for the anti-encoding to ensure the most significant contrast and applied the corrective mapping to restore the anti-category embedding to the true embedding distribution. The full pipeline for how this can be done is shown in Figure 1.

The inference was then done by executing on the prior text, recurrently, in X_{prompt} by sampling from the latent token probabilities given by the following formula, while α is a scaling value for the strength of the control methodology. z_{base} is the latent token probabilities output by some independent language model with input of X_{prompt}, which we wish to provide the control for.

$$z_{T^+} = M_E(\hat{h}_{T^+}, X_{prompt}) \tag{7}$$

$$z_{T^-} = M_E(\hat{h}_{T^-}, X_{prompt}) \tag{8}$$

$$z_{fin} = z_{base} + \alpha(z_{T^+} - z_{T^-}) \tag{9}$$

For our evaluations, we utilized a beam search methodology with a beam size of 10.

5 Training

Based on initial testing, we went with unreduced embeddings extracted from the T5-Large encoder. The unreduced embeddings performed best in producing well-balanced SOMs with low average quantization error and a strong mean number of samples quantizing to each SOM node. The embeddings were extracted by concatenating the categories from our dataset and taking the mean over all tokens for the final hidden layer of the encoder.

For the SOM we utilized a near approximation of a (4,4) Goldberg Polyhedron with a total of 488 nodes. We utilized an exponentially decaying learning rate as well as a Gaussian scaling kernel function to determine value adjustment for nearby weights. The Gaussian scaling kernel possessed an exponentially decaying standard deviation to fine-tune for best quantization in later epochs. We trained for 40 epochs beyond which little benefit

Model	Perplexity	ROUGE-1			ROUGE-2			ROUGE-l			Bleu	
		R	P	F	R	P	F	R	P	F	1	2
GPT2												
w Prompting	3.74	.21	.60	**.29**	**.13**	.40	**.18**	.20	.59	**.28**	.20	.10
w/o Prompting	3.30	**.23**	.48	**.29**	**.13**	.30	.16	**.22**	.47	**.28**	**.22**	**.11**
GPT2 w/ (+)												
$\alpha = 0.2$	3.279	**.23**	.48	**.29**	**.13**	.30	.17	**.22**	.47	**.28**	**.22**	**.11**
$\alpha = 0.4$	**3.309**	.22	.50	.28	**.13**	.32	.17	.21	.49	**.28**	.21	**.11**
$\alpha = 0.7$	3.308	.21	.55	.29	**.13**	.36	.17	.21	.54	**.28**	.20	.10
$\alpha = 1.0$	3.317	.20	.58	.28	.12	.39	.17	.20	.58	**.28**	.19	.10
$\alpha = 1.5$	3.330	.20	.63	.28	.12	.43	**.18**	.19	.62	**.28**	.17	.10
$\alpha = 2.0$	3.344	.19	**.66**	.28	.12	**.45**	**.18**	.19	**.65**	**.28**	.17	.10
GPT2 w/ (+&-)												
$\alpha = 0.2$	3.291	**.23**	.48	.28	**.13**	.30	.17	**.22**	.47	**.28**	**.22**	**.11**
$\alpha = 0.4$	3.292	**.23**	.48	.28	**.13**	.30	.16	**.22**	.47	**.28**	**.22**	**.11**
$\alpha = 0.7$	3.293	**.23**	.48	.28	**.13**	.30	.16	**.22**	.47	**.28**	**.22**	**.11**
$\alpha = 1.0$	3.293	**.23**	.48	.28	**.13**	.30	.16	**.22**	.47	**.28**	**.22**	**.11**
$\alpha = 1.5$	3.275	**.23**	.48	.28	**.13**	.30	.16	**.22**	.47	.27	.21	**.11**
$\alpha = 2.0$	3.275	**.22**	.48	.28	**.13**	.30	.16	**.22**	.47	.27	.21	**.11**

Table 1: Results of our various model architecture and baselines with a range of model parameters on 500 samples drawn from our Wikipedia test set. In the above table (+) stands for our on-category expert while (+&-) stands for use of both contrastive experts. α is the expert strength parameter.

was seen. Distance in our scaling kernel function was parameterized as arc angle between two nodes to provide a smooth and linear representation for the training.

For our expert language model, we utilized independent adapters between each GPTBlock with a reduced dimensionality of 256. Classifier-free guidance and token masking chances were set to 20% each. The model was allowed to fine-tune the GPTBlocks and LM head and was done with an AdamW optimizer and linear learning rate schedule. A max of 128 tokens was allowed for training, using truncation and padding. A contrastive loss value of 0.9 was utilized, and we sample the true opposite category embedding with a standard deviation of 0.1 (in radians). Our expert model was unable to be trained until convergence in the training time allowed (24 hours).

All models were trained on a commodity RTX 3080 12GB with the maximum batch size allowed.

6 Results

Here we will go over the result of our model. We were unable to achieve convincing benefits from our proposed methodology, which we will discuss after presenting the details.

6.1 Baselines

There are few baselines we can utilize for our project as no other model we are aware of can take open-ended auxiliary prompts for conditioned control of text generation. As such, we choose to compare our model against GPT2 on its own, GPT plus our primary expert, and GPT with a pre-prompt comprising the category labels added to the input text.

6.2 Evaluation Metrics

Outside of a qualitative evaluation of the topical alignment and generation fluency, we are unable to locate any task-specific evaluation metrics that would be optimal for our project. We have instead come up with a series of evaluations we will perform to judge task performance which we discuss below. To test how our model handles out-of-domain prompts we have utilized OpenAI's ChatGPT to generate a series of test prompts and to label each prompt with corresponding categories, which we spot-checked for reasonability. We use this additional test set in addition to our Wikipedia-drawn test set.

Model	Perplexity	ROUGE-1			ROUGE-2			ROUGE-l			Bleu	
		R	P	F	R	P	F	R	P	F	1	2
GPT2												
w Prompting	**3.088**	.56	**.88**	**.67**	.50	**.72**	**.58**	.56	**.88**	**.67**	**.15**	**.13**
w/o Prompting	3.272	**.61**	.39	.45	**.53**	.30	.36	**.61**	.39	.45	.18	.14
GPT2 w/ (+)												
$\alpha = 0.2$	3.290	**.61**	.39	.45	**.53**	.30	.36	.60	.39	.45	.18	.14
$\alpha = 0.4$	3.317	**.61**	.44	.48	**.53**	.33	.38	.60	.44	.48	.19	.15
$\alpha = 0.7$	3.223	.59	.55	.54	.52	.42	.43	.59	.54	.54	.19	.15
$\alpha = 1.0$	3.227	.60	.58	.56	.51	.45	.45	.59	.58	.56	.20	.16
$\alpha = 1.5$	3.221	.58	.63	.59	.51	.50	.48	.58	.63	.58	.21	.17
$\alpha = 2.0$	3.342	.58	.64	.59	.51	.50	.48	.58	.64	.59	.19	.15
GPT2 w/ (+&-)												
$\alpha = 0.2$	3.272	**.61**	.39	.45	**.53**	.30	.36	**.61**	.39	.45	.18	.14
$\alpha = 0.4$	3.273	**.61**	.40	.45	**.53**	.30	.36	**.61**	.40	.45	.18	.14
$\alpha = 0.7$	3.282	**.61**	.39	.45	**.53**	.30	.36	**.61**	.39	.45	.18	.14
$\alpha = 1.0$	3.286	**.61**	.40	.46	**.53**	.30	.36	**.61**	.39	.45	.18	.14
$\alpha = 1.5$	3.303	**.61**	.39	.45	**.53**	.30	.36	**.61**	.39	.45	.18	.14
$\alpha = 2.0$	3.312	**.61**	.39	.45	**.53**	.29	.35	**.60**	.39	.45	.18	.14

Table 2: Results of our various model architecture and baselines with a range of model parameters on 200+ samples drawn from ChatGPT generated test set. In the above table (+) stands for our on-category expert while (+&-) stands for use of both contrastive experts. α is the expert strength parameter.

6.2.1 Reference Based Language Metrics

We will compute reference-based language metrics (BLUE and ROUGE) between our generated results and the ground truth Wikipedia summaries. An issue with these metrics will be that the ground truth text will be very stylistic (Wikipedia summaries). We show, however, that the generation results will be more style agnostic, as the DExperts work theorized the contrastive nature of the expert models works to cancel out stylistic effects. We can verify if stylistic effects are present through comparisons of our two test sets.

6.2.2 Distributional Metrics

As a test of fluency, we will measure the perplexity of the outputs with GPT2-Large. This will not be an ideal metric, as some level of distributional shift is desired. However, very drastic perplexity values may indicate a fluency issue.

6.3 Quantitative Results

Our results are shown in Table 1 for our Wikipedia test set and in Table 2 for the ChatGPT generated out-of-domain test set. As can be seen, our methods when using both positive and negative experts (contrastive formulation) largely achieve consistent results across all tested values of α. Further, these results are essentially the same as our baseline testing when done without any category pre-prompting. This means that our expert language models are failing to differentiate based on the category conditioning embeddings, and so the two experts simply cancel each other out in the current state.

We observe improvement in results for ROUGE scores precision and F1 values when only using the positive expert model conditioned on the category embeddings compared to the GPT2 baseline by itself; we likely observe further increases as the expert model weight parameters are further increased. In fact, we see this effect across both our in-domain and out-of-domain test sets, implying that this is not simply the fact that our expert model is fine-tuned to produce the stylistic text of our Wikipedia training data. Instead, it does appear to be obtaining additional useful conditioning from the category embeddings.

We cannot rule out some of the improvement with only the positive expert from stylistic effects, however. As is seen against our GPT2 baseline with the categories appended to the beginning of the prompt, the baseline significantly exceeds our positive expert on out-of-domain data and manages to make significant steps to closing the performance gap even on in-domain data.

7 Discussion and Future Directions

The results we see for our contrastive topic control approach have multiple possible explanations. The first is the possibility that our category embeddings are not information rich enough to provide consistent contrastive signals to our expert language model, which may be capable of being resolved through an alternative encoding strategy. Despite trying multiple recent and high-performance text encoding models, we were limited to a small set of different methods for obtaining the embeddings from the encoder's hidden representations due to compute time limitations. Alternative options such as using the CLS token feature or even larger versions of the encoders could have produced better results potentially.

It is evident from a qualitative evaluation of the text generation and inspection of the training curves that our model fails to strongly differentiate text generated with conditioning on the primary embedding as opposed to the anti-embedding. This is despite the training loop possessing a contrastive loss component. Unfortunately, higher values of contrastive loss (such as 0.6 or 0.5) did produce a high distributional differential, but this was done by shifting the anti-embedding results to a very unrealistic probability distribution for the actual text. For example, it often just produced repeated sequences of special tokens as opposed to the actual text. Revisions of the loss function to both promote contrast while also enforcing that anti-embedding generation results fit a realistic distribution may be a future direction.

Finally, it is possible the dataset quality and size is a limiting factor. While we believe Wikipedia summaries are well described by the obtained categories, this may not be the case from a textual perspective. Our positive expert-only results do indicate some level of stylistic overfitting. It is also possible the SOM and the category embeddings we are fitting them to do not describe the proper features we need for our purposes. If our text encoder is extracting some deeper meaning from the categories but does not reflect information on the type of language that would be used in discussing those categories, our results may be affected as a result. These are all areas we would need to explore in order to take this project further.

8 Conclusion

In conclusion, we propose a framework for diverse and open-ended topic control which can be applied to any shared vocabulary large language model without further pre-training. We have broken the solution into a multi-step pipeline with independently verifiable models produced in each stage.

Our results were largely reliant on, and as such limited by, two factors: 1) meaningful category embeddings for our textual category labels and 2) meaningful inversion properties from our enforced SOM structure. We were unable to show conclusively that our approach has the potential to produce a successfully controlled generation of text conditioned on a given topic descriptor or increases topic stability in a long-form generation but believe it is a valid step in the correct direction for doing so. Future work can build on the dataset, codebase, and findings we obtained to refine further controllability attempts.

References

Paweł Budzianowski and Ivan Vulic. 2019. Hello, it's gpt-2 - how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. In *EMNLP*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Teuvo Kohonen. 1990. The self-organizing map. *Proc. IEEE*, 78:1464–1480.

Teuvo Kohonen. 2004. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith,

and Yejin Choi. 2021a. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *ACL*.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. 2021b. Dexperts: Decoding-time controlled text generation with experts and anti-experts. *arXiv preprint arXiv:2105.03023*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Jianmo Ni, Gustavo Hern'andez 'Abrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Matthew Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *FINDINGS*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "transforming" delete, retrieve, generate approach for controlled text style transfer. 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 3269–3279, Hong Kong, China. Association for Computational Linguistics.

Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. 2021. Controllable generation from pre-trained language models via inverse prompting. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.