Towards Generalized Control: On-the-Fly In-Topic Generation

Michael Tang[1] **Jiatong Yu**[1]
[1]Department of Computer Science, Princeton University
{mwtang, jiatongy}@princeton.edu

Abstract

In this work, we propose the novel on-the-fly in-topic generation task to extend in-topic generation to unseen, general-purpose topics. Towards this end, we motivate and build a benchmark involving news article generation with article titles as control, and develop various models to tackle this task by leveraging prompting, retrieval, and inference-time topic modeling. We find that that building on-the-fly Bag-of-Words (BoW) models and leveraging latent space modification techniques like PPLM [3] is a promising method for this new kind of fine-grained in-topic control, although zero-shot prompting of Large Language Models remains a strong baseline, whose limitations we explore. Finally, we propose various automated evaluation metrics for our task based on sparse and dense TF-IDF and SimCSE [4] encodings, and show that they behave similarly to human scores for in-topicness, opening up new promise for evaluations of control that go beyond human annotations.

1 Introduction

Even amid the advent of improvements for using pretrained Large Language Models (LLMs) for text generation tasks (e.g. through chain-of-thought reasoning [8] or optimized prompting), LLMs infamously suffer from alignment problems such as hallucinations, toxicity, or otherwise repetitive, off-topic, and/or degenerate outputs.

Controlled text generation (CTG) has the promise of both being a generalized way for models to plan at a high level as well as be functionally robust, aligned, and steerable. Unfortunately, the current state of the field is:

1. *Fragmented.* Papers generally focus on disjoint generation tasks with disjoint task-specific approaches, often highly task-specific (e.g. couplet generation, prevention of toxic generation)
2. *Inaccessible.* The strongest results come from expensive fine-tuning techniques, such as the usage of human demonstrations and human sample rankings in InstructGPT [5] for two rounds of supervised and RLHF (Reinforcement Learning with Human Feedback) fine-tuning, respectively
3. *Difficult to evaluate.* Aside from task-specific scenarios (e.g. evaluating whether a generated couplet rhymes according to a phonology dictionary), past work exclusively relies on human evaluation, which is both expensive and difficult to scale. As a side effect, this worsens the impediments to replication and cross-work comparisons, as works must spend precious human annotations evaluating other models in order to make a comparison, especially in the case of fragmented tasks and even within-task datasets whose ground truth annotations are thus incomparable

One of the more common CTG tasks is in-topic generation [9] [3], where models are scored on their language modeling as well as an auxillary objective of staying "on-topic," as judged by human annotators, where the topic is usually selected a priori from a small set of well-defined themes. For example, the list from a recent work [3] is fantasy, space, politics, military, religion, computers, and legal. We aim to relax these hard-coded topic definitions in hopes of taking a first step towards a unifying framework for controlled-text generation.

2 Related Work: PPLM and FUDGE

Two of the leading prior works in in-topic generation are Plug and Play Language Models (PPLM) [3] and FUDGE: Controlled Text Generation With Future Discriminators [9]. Both works involve training a sepa-

rate attribute classifier — for topic control, this is a classifier for a given set of topics based on the currently generated tokens. In PPLM, for each generated token, this loss is used to compute a gradient update to shift the latent distribution of the LM towards the topic, as well as provide a score for ranking LM samples based on how on-topic they are. In Fudge, the classifier's log probabilities are simply added to the LM output probabilities to guide the generation. PPLM and Fudge collectively try (1) a Bag of Words (BoW) classifier with a manually selected word list per category, (2) a single-layer linear classifier, and (3) an LSTM-based classifier.

As stated in the introduction, a common theme between these approaches is weak evaluation for on-topicness: PPLM only uses human evaluation, and Fudge uses (alongside human evaluation) an ad hoc success metric defined as the average number of distinct words in a heldout bag constructed by compiling nontrivial GloVe-similar words to the bag used in the attribute classifier. It is easy to see that this lends itself to a superficial on-topicness, i.e. as long as the perplexity and grammaticality is generally preserved (this is evaluated using other automated metrics) the model is only incentivized to mention, rather than fully involve, words that are related to the topic. In particular, CTG techniques that overfit highly to the attribute classification models (e.g. BoW) can perform well by injecting in-topic words in places where they grammatically make sense, without actually doing any reasoning or planning on what the domain represents. It is also evident that this leads to artifacts such as certain types of in-topics generations (e.g. lists) with a high density of distinct in-topic words being favored over more common types of in-topic generations.

2.1 Retrieval: SimCSE

One of the approaches for our task involves retrieving other in-topic text from a given corpus to supplement a given example to provide a more well-defined and complete view of the topic (see 2).

Our choice of retriever is SimCSE [4], a dense encoding model composed of BERT embeddings fine-tuned with a supervised contrastive objective from NLI datasets with entailment treated as positive pairs and contradiction treated as hard negatives:

$$l_i = -\,log\,\frac{e^{sim(h_i,h_i^+)}}{\sum_j e^{sim(h_i,h_j^+)}}$$

It captures semantic similarity especially well, and we specifically use its retriever functionality by encoding a query (article title) along with each candidate (sentences from other articles) where query similarity is defined as embedding cosine similarity. We found that SimCSE similarity is surprisingly invariant to length, a fact we leveraged in using it to retrieve full texts using titles, or words using sentences, etc.

3 The On-the-Fly In-Topic (OutFIT) Generation Task

3.1 Motivation

We generalize in-topic generation on a small finite set of fixed topics to handle a *continuous topic space*. Concretely, we will construct a task where instead of a unified topic (say, science) with a large dataset and existing in-topic attribute classifier that we want to train a model to generate in, we want a model that can adapt to any topic presented at inference time, which we call an *on-the-fly topic*.

To ground the benchmark in a real-world task while preserving generality, we model in-topicness by the title-article relation in news articles where the body text is described at a high level by the title information, but they are thematically different and do not follow by typical language semantics. In particular, we might be interested in generating article body text conditioned on both the title and a snippet (e.g. the first few words or sentence of the true body text), which play different roles. Generally, we see the title as providing a high-level plan, whereas the snippet sets a specific starting point, which may only encompass a small detail of the larger story.

Importantly, we can construct motivating examples where neither are sufficient to solve the task, e.g.:

1. Title: "Fusion to Replace Geothermal Energy by 2050." Snippet: "Yesterday, the president stopped by Lawrence Livermore laboratories to congratulate the growth lead". The title contains the critial geothermal energy part of the story, whereas the snippet includes the important fact that the starting anecdote must be able an interaction between the president and the lab growth lead, neither of which is nonobvious.
2. Title: "Chip Shortage Has No End In Sight, Military Operations to Blame." Snippet: "Snack manufacters bemoan the ongoing overconsumption

of Lay's potato chips." The title is ambiguous with respect to whether computer or potato chips are the subject, and the snippet says nothing about the second piece on military operations being the cause.

Intuitively, the high-level title and low-level snippet complement each other, and reflect the planning-execution dynamic of human conversation.

3.2 Task specification

Formally, the task is as follows: given an article title string x_t, description string x_d (can be seen as a longer-form title specification), a snippet string x_s composed of the first five words of the article, produce a generation string of the article body text $\hat{y}$ that is similar to the ground truth y.

3.3 Task variants: open-book and open-library

We recognize that in the absence of existing datasets, classifiers, or even bags of words for a fixed topic, along with the strong thematic cohesiveness of the canonically chosen topics (e.g. military, fantasy, compared to something as specific as "police investigation in Florida" in our task), an on-the-fly topic may be underspecified by just conditioning on the short article title itself. Thus, we consider relaxations of the task that allow the model more information in the form of the text of other articles to references, motivated by the way that human writers usually study in-topic and in-style reference texts or past papers for a given topic before writing their own work:

1. **Closed-book.** This is just the base formulation above, where x_t is the title string for the desired article we want to generate.

2. **Open-book.** We provide the full text of k related articles, which we will denote *references*, $\mathbf{z} = z_1, ..., z_k$ (topics are similar as judged by a retriever model, see 3.3), along with the title of the given article x_t

3. **Open-library.** Same as the open-book setting except retrieval is do-it-yourself. (intuition is that integrating the knowledge of how to retrieve and compare texts may help the in-topic generator itself) Explicitly, we provide the full texts of a large *library* of related articles Z ($y \notin Z$), along with the title of the given article x_t. The expectation is that at inference time, the most efficient and effective thing for the model to do is first retrieve the most relevant full texts, which turns this into a harder version of the open-book setting.

In general, let $x = (x_t, x_d, x_s, \mathbf{z})$ where $\mathbf{z}$ may be a corpus Z, a vector of references, or empty depending on the variant. Then, the task can be concisely described as predicting y given x.

Remark on open-library. We recognize that although our approaches are focused on inference-time control, other attempts at solving this task may not respect this expectation, e.g. a model may attempt to train on the data in a few examples of Z at the beginning and amortize the time by using this knowledge across many subsequent examples. However, a simple change where we take disjoint subdatasets with different writing styles for different examples, and randomize the order, ensures that for two examples with $(Z_i, ..., y_i)$ and $(Z_j, ..., y_j)$, Z_i is no more helpful for deducing y_j than any other information used for training outside the task.

We focus on the open-library and closed-book settings, evaluating PPLM-based methods as a baseline for open-library and zero-shot prompting as a baseline for closed-book.

Retrieval details. Suppose a given retrieval method (for the open-book case we choose SimCSE) is a function $f(q, Z)$ between a query q (e.g. title) and a corpus Z. Then we fix task-specified parameters θ for the min threshold for considering a candidate z_i a reference, and k for the desired number of references. Then to get the open-book references for a title x_t, we pick, as canonical, $\text{argmin}_{z_1 \neq ... \neq z_k} \sum_{i=1}^{k} f(q, z_i)$ between the title and texts, and then return the $\leq k$ texts among those that have $f(q, z_i) \geq \theta$. When constructing the task dataset, we ensure all examples have k references, which can be seen as quality filtering in the topic space (i.e. well-defined, common topics). Note that for open-book Z is the entire training dataset.

Note that our PPLM baselines for open-library use this exact setup for reference retrieval from a given Z as part of the model pipeline, except with θ, q as tunable hyperparameters and no guarantee that we can disregard an evaluation example if we get $< k$ θ-similar results. We evaluate only Z as the entire training dataset in those cases (for training, the we remove the text of the given article), but future work may want to explore more limited or specific corpuses in the open-library setting.

3.4 Dataset

We use the Common Crawl News dataset [1], which contains articles with titles, short descriptions, body text, and metadata, and which is among the largest and most complete news datasets available. For quality purposes, we filter out articles with domains other than https and www, as well as those with short titles (≤ 30 chars) and descriptions (≤ 60 chars).

In this setup, each data instance is composed of a title, a summary, and a text body. Our preliminary exploration of the CC News dataset shows that some title does not sufficiently capture the topic of the news article, mostly due to brevity. Therefore we concatenate title and truncated summary for each data instance as the topic description, and consider the text body a ground truth instance.

3.5 Task evaluation

The evaluation metrics developed in recent controlled generation tasks are not sufficient in the topic-agnostic setting. FUDGE[9] uses a pre-defined Bag of Words (BoW) model for each topic class, which is not applicable to agnostic in-topic generation task, where we do not assume any prior knowledge or constraints on the topic. PPLM[3] uses only human annotations. This naturally raises a question of whether there exists a topic-agnostic automated evaluation metric that, given a topic description, a ground truth instance, and a model generation, measure how well the generation follows the given topic description.

We propose two evaluation metrics under this framework.

Our first metric uses a weighted Bag of Word approach called *TF-IDF*, short for Term Frequency - Inverse Document Frequency. At a high level, *TF-IDF* computes the intersection between two weighted vocabulary distributions given a generation and a ground truth instance. To offset common words such as "and", "a", "the", etc. which does not contribute to our measure of in-topic degree, the algorithm weights each word by the inverse of its frequency. If a word occurs rarely in other news articles but frequently occur in a particular instance, it is highly likely that the word is highly related to the topic and therefore could be used in evaluating generations.

As introduced in section 2.1, Gao et al.[4] developed a contrastive-learning based encoding model that embeds sentences into vectors. Given the assumption that the news article itself – the ground truth – aligns with the topic perfectly, we can measure how "in-topic" a generation is by calculating its semantic distance to the ground truth embedding. As our later experiments suggest, the sentence embedding approach aligns with human annotation the best.

Remark. One caveat is that the aforementioned evaluation metrics do not measure semantic coherence.For instance, if the model generates only a few random symbols after the given prompt, and if we feed the prompt concatenated with {generation, ground truth} pair to SimCSE encoder model, the cosine similarity will be very high. Therefore we only consider the bare generation and body text for a topic (without prompt or title information) for any of the aforementioned evaluation metrics.

To measure both in-topic degree and semantic coherence, a weighted interpolation between our evaluation metrics and LLM perplexity measurement could potentially be a better metric, which could be considered for future work.

4 Task Baselines

We are mostly interested in inference-time approaches towards CTG in our baselines, as opposed to expensive fine-tuning approaches (e.g. train a model to accept on-the-fly information through a separate specially designed channel), as they are more lightweight, versatile, and build off of advances of foundation language models.

4.1 Zero-shot learning

More recently, in-context learning has promised a way to use few to no demonstrations to achieve state-of-the-art generalization, especially in LLMs. Thus, as a baseline for OuTFIT across different model scales, we evaluate various sizes of GPT-2 as well as SoTA LLMs GPT-3 [2], OPT [11], as well as the context-specific science language model Galatica [7] in a zero-shot setting.

Prompting pretrained LLMs for in-topic generation is a natural first-pass baseline. We construct the prompt as follows.

- Template: "Generate a long article given the title and summary below. Title: [INPUT], Summary [INPUT], Generation:"
- Due to the constraint of context window, we only

append the first sentence in the summary into the template. When the sentence is still too long, we truncate them to the maximum window size.

We run this prompting scheme on the GPT2, OPT, and Galactica families, with results in Table 4.

4.2 On-the-fly PPLM (OPPLM)

Since PPLM is designed to be modular and have its attribute classifier switched out between tasks, we are interested in to what extent this modularity extends to a topic that the model only learns about on-the-fly. This takes two forms:

4.2.1 Topic discriminator

In the open-library setting, we use the given topic information $x_t, \mathbf{z}$ to create dataset and train a topic discriminator on-the-fly.

1. Use the title as a query to retrieve a set of full text references $\mathbf{z}$ with high similarity to the title
2. Split references into groups of c sentences, and give them label 1 for on-topic (we empirically find $c = 1$ works the best)
3. Also sample random texts from the corpus and split them into groups of c sentences to get negative examples, which have label 0
4. Train the in-topicness classifier for m epochs, where m is a hyperparameter
5. Use the classifier as the attribute model for PPLM

The classifier architecture here is a single-layer MLP, as in the original PPLM paper.

We will denote this model by OPPLM-D (on-the-fly PPLM + discriminator).

4.2.2 Topic bag of words

1. Use the title as a query to retrieve a set of full text references $\mathbf{z}$ with high similarity to the title
2. Use a BERT model fine-tuned for keyword extraction to extract keywords for each reference (pretrained model from HuggingFace [10])
3. Filter the resulting keywords by their SimCSE similarity to the original title using a hyperparameter threshold θ'
4. Use the resulting words as the Bag of Words model used for PPLM (detailed computation for getting the gradients from the BoW can be found in [3])

We will denote this model by OPPLM-BoW (on-the-fly PPLM + BoW).

Note that the base language model on top of which we apply PPLM-based techniques is GPT-2 medium [6], following the original paper [3].

5 Results

The main results are summarized in 1. Overall, larger models generally outperform the smaller ones in terms of both SimCSE and TF-IDF Score, with the best results obtained by OPT-1.3B. We also note that the distribution match matters significantly for in-topic control just as it does with general language modeling: Galactica performs poorly, in fact significantly worse than GPT2-large despite its massive size, due to it being fine-tuned on scientific writing.

This also suggests that conversely a news-specific fine tuned model may perform somewhat better on the task overall due to the dataset choice, so a possible direction for future work might look at how on-the-fly improvements can lessen this gap, or generally how inference time control can make up for sometimes unavoidable distributional differences from pre-training.

OPPLM-D performs similarly to base GPT2-medium, and OPPLM-BoW outperforms it but still falls short of larger models using zero-shot prompting.

Remark. All PPLM-based experiments were run on extremely limited datasets due to their very long inference time (≥ 1 min per example for the base PPLM and multiple mins per example for on-the-fly discriminators), as well as the additional need to tune highly sensitive hyperparameters (step size, KL scale, and GM scale, with details in the paper [3]) relating to the tradeoff between the language modeling and control objectives which have new optima for each variation of the baseline. While we initially selected PPLM as a control method due to its lightweight architecture building off of an existing model, its inference time costs turned out to outweigh possible training cost amortized over inference runs. This is an important observation to keep in mind when building related LLM infrastructure (e.g. prompt tuning, calibration, etc), especially any that run computations on intermediate layers or logits that quickly scale with the LLM

Model	SimCSE Score	TF-IDF Score
GPT2-medium	0.01	0.06
GPT2-large	0.40	0.19
GPT2-XL	0.45	0.23
OPT-1.3B	**0.52**	**0.35**
Galactica-1.3B	0.26	0.04
OPPLM-D + GPT2-medium	0.07	0.04
OPPLM-BoW + GPT2-medium	0.21	0.09

Figure 1: Main OuTFIT Results

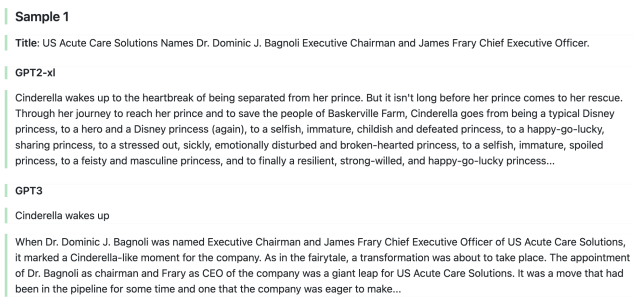

Figure 2: Sample generation 1 of GPT2-XL vs. GPT3 with adversarial prompt ("Cinderella wakes up")

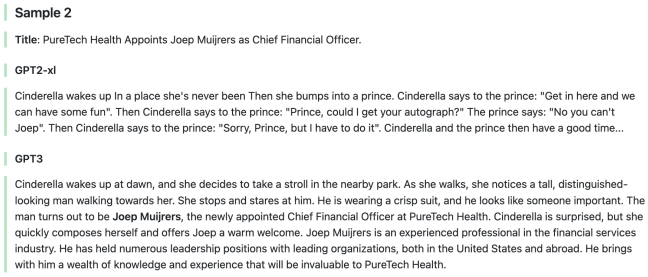

Figure 3: Sample generation 2 of GPT2-XL vs. GPT3 with adversarial prompt ("Cinderella wakes up")

size beyond what we faced here with GPT-2.

6 Discussion

6.1 Adversarial prompts

To what extent does prompting suffice? We ablate the zero-shot prompting baselines by using handwritten adversarial prompts (randomly picked among "Cinderella wakes up," "The potato wakes up," "The chinchilla wakes") after the title that have nothing to do with the article. Below are two examples to show that GPT-3 has a much stronger understanding of the prompt and can follow the instruction of the prompt, while even GPT2-XL falls to the adversarial prompts.

We generated 500 samples of this adversarial scheme, and the results are shown in combination with the next section.

6.2 Helpful prompts

This ablation is a mirror of the adversarial prompts. Here, the snippet is the truncated first sentence of the ground truth article text. The combined result with adversarial prompting is shown in Table 4, where we denote models that ran with adversarial prompts as (Adv), and models that ran with helpful prompts as (Help).

Across model scales, adversarial and helpful snippets have a significant effect on in-topic performance. The fact that a snippet concatenated to the topic instruction can boost SimCSE score up to 49% suggests that the pre-trained LLMs, at least in the case of GPT2-Large and GPT2-XL, are not truly learning the topic instruction in the prompt beyond surface level conditioning. As shown qualitatively in Figure 2 and Figure 3, GPT3 is much more robust, having the emergent property of learning and remembering the topic description, and keeping itself aligned with the instruction along the generation.

6.3 OPPLM: Discriminator vs. BoW

We find that OPPLM-D performs significantly worse than OPPLM-BoW, even though the a priori discriminators and BoW attribute models performed similarly in the original PPLM work [3]. Intuitively, under open-library, both models should access to similar in-

Model	SimCSE Score	TF-IDF Score	BLEU Score
GPT2-large (Adv)	0.27	0.16	0.03
GPT2-XL (Adv)	0.29	0.19	0.06
GPT2-large	0.40	0.19	0.08
GPT2-XL	0.45	0.23	0.08
GPT2-large (Help)	0.76	0.47	0.06
GPT2-XL (Help)	0.77	0.51	0.06

Figure 4: Adversarial and Helpful Prompting Results

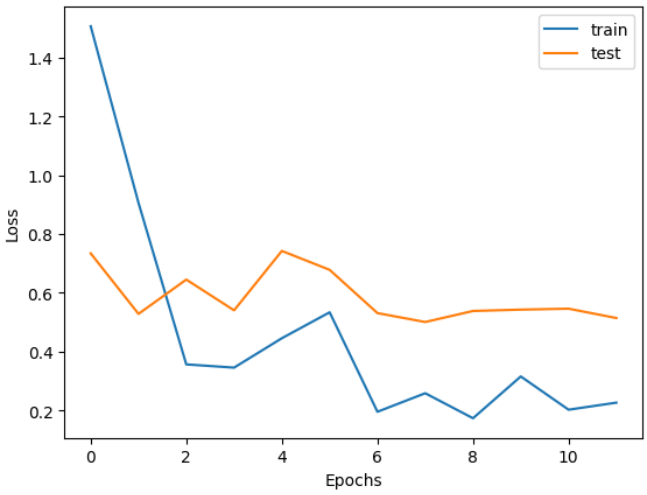

Figure 5: Sample on-the-fly training process with ad hoc test loss superimposed

formation, so is it the case the OPPLM-BoW is just a more natural way of encoding it?

To investigate this, we evaluate some of the on-the-fly classifiers trained under OPPLM-D by constructing small test sets composed of c-sentence groups from the ground truth article text as positive examples and random c-sentence groups from the training dataset as negative examples, which complement the on-the-fly training sets. In the figure 5, we show a qualitative example of a training curve with this ad hoc test loss superimposed. We find that the classifiers are generally not very robust, with test accuracy often not significantly above 50% and even high-test-accuracy classifiers falling quickly to in-distribution adversarial inputs (e.g. we can change a handful of words to their synonyms retaining news writing style, and find that the probability changes drastically), demonstrating instability. Overall, we find that the linear MLP discriminator with the on-the-fly datasets is not sufficiently expressive to define the boundaries of the

Title: Acclaimed ensemble Solomon's Knot to sing Bach at Nottingham church
Bag: ['solomon', 'gospel', 'baritone', 'chorus', 'musica', 'tenor', 'soprano', 'choral', 'bach', 'handel', 'christian', 'alto']

Title: North Korea remains United States' most imminent threat: outgoing Pacific Command chief
Bag: ['north', 'pacific', 'ambassador', 'korea', 'command']

Title: Nigeria World Cup 2018 squad and team guide
Bag: ['fifa', 'africa', 'gabon', 'nigeria', 'ghana', 'nations', '2017', 'world']

Figure 6: Sample bags of words

given topic. While larger LSTM-based or transformer-based discriminators are possible, we found these impractical to train on-the-fly.

In contrast, we find that the the OPPLM-BoW method results in bags of words that accurately capture the topic implied by article titles. From qualitative examination 6, we find that a combination of retrieval and keyword extraction is able to produce on-the-fly bags of words that succinctly capture the topic implied by the title, as well as surface other words that build a more complete picture of it. (e.g. the notion of singing Bach at a church is made more complete by the surfacing of words like "choral, "soprano," and "tenor" which do not necessarily appear in the title but are likely to be relevant in the ground truth article text)

The advantage of PPLM-based methods is particularly evident in adversarial prompting settings as shown in 7. We see that while the base GPT2-medium conditions heavily on the language modeling surrounding the adversarial snippet, the PPLM-based model provides continuous conditioning via the latent space updates to bias towards words related to the topic.

However, we note that most of the PPLM contributions are limited to the words in the bag, showing a surface-level understanding of the topic. For future work, this motivates a BoW-type construction with higher-level, perhaps phrasal, semantic embeddings as a promising direction for tackling OuTFIT.

Title: Acclaimed ensemble Solomon's Knot to sing Bach at Nottingham church

GPT2-medium
The potato wakes up. The cat goes back to sleep. The dog is asleep. The horse is awake. The horse and cat sit together. The cat and dog lie side by side. In a garden. In the morning the horse goes to his room. The cat

PPLM-BoW
The potato wakes up the sun, but Solomon, his wife and their chorus are ready with an all-star chorus. The band is led by composer John Hammond (1878-1947) and conductor Richard Houghton. Solomon's musical score includes the song's

Bag: ['solomon', 'gospel', 'baritone', 'chorus', 'musica', 'tenor', 'soprano', 'choral', 'bach', 'handel', 'christian', 'alto']

Figure 7: Sample generation of GPT2-medium vs. OPPLM-BoW on an adversarial prompt ("The potato wakes up")

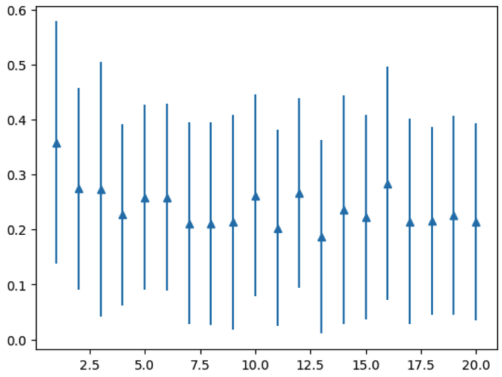

Figure 8: Sentence-by-sentence degeneration: sentence-level SimCSE score by sentence #

6.4 Sentence-by-sentence degeneration

Given the autoregressive property, we expect that pretrained langauge models may "forgetting" a topic under zero-shot prompting as longer texts are generated. We concretely evaluate this phenomenon by running the SimCSE metric on $\langle s, y \rangle$ pairs for each sentence $s \in \hat{y}$ in the generated text and the actual article text y. We randomly sampled 100 generations with more than 15 sentences and ran the SimCSE and TF-IDF metrics on each data instance, and obtained the following results 8 and 9. Although not statistically significant, the mean generally decreases, supporting the notion that degeneration is a valid concern, although likely a very noisy phenomenon to analyze.

Overall, this further motivates designing models specific to the controlled generation task, which are theoretically length-invariant (although this empirically has not yet been achieved with PPLM due to degeneration).

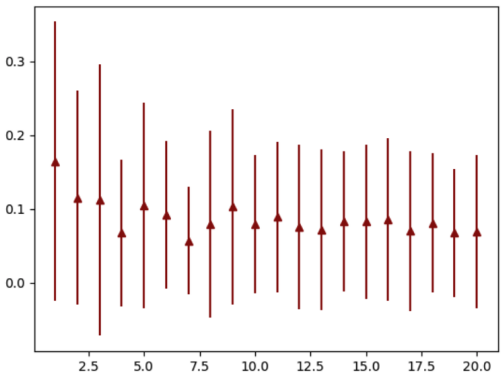

Figure 9: Sentence-by-sentence degeneration: sentence-level TF-IDF score by sentence #

6.5 TF-IDF and SimCSE scores are similar to human ratings

To measure how accurate our TF-IDF and SimCSE metrics are with respect to human evaluation for OuTFIT, we collected human annotations on a 0-10 Likert-style scale regarding the question "How well does this article stay on topic?" on the tail ends of GPT2-XL generations, which exhibit a particularly large range of on-topicness, and computed Pearson and Spearman correlations between the TF-IDF scores and human ratings of the tail-end generations.

Metric	Pearson (p-value)	Spearman (p-value)
TF-IDF	0.482 (0.03)	0.446(0.048)
SimCSE	**0.608** (0.004)	**0.675** (0.001)

We found that both metrics are fairly correlated with the human ratings, with SimCSE slightly more so, which indicates it may be a better candidate for automated evaluation. Note that all p-values are below 0.05. We also see that the Spearman and Pearson correlations are similar, implying a high degree of linearity in the correlation, as expected.

7 Conclusion

We are generally interested in a more unified approach towards controlled text generation, and make early strides towards this by proposing a novel relaxation of in-topic generation, where instead of considering an a priori fixed set of topics, the model must adapt to a topic on-the-fly for each example, and must be able to adapt amid a large topic space. Concretely, we build the OuTFIT benchmark based

on news articles with titles as control, and explore a prompting-based baseline for it, as well as an alternative approach focused on retrieving related articles and building an attribute model on-the-fly to use with inference-time control methods like PPLM. We also propose two automated metrics for in-topic generation, TF-IDF Score and SimCSE Score, and find that both correlate reasonably well with human evaluation, suggesting that similar techniques may be used for automated evaluation of control in general.

8 Acknowledgements

We would like to thank Professor Danqi Chen and Alexander Wettig for their kind feedback and support, as well as Tianyu Gao, Shunyu Yao, and Howard Chen for their feedback in early discussions.

References

[1] Vladimir Blagojevic and Julien Nioche. *Common Crawl News Dataset*. 2016.

[2] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. DOI: 10.48550/ARXIV.2005.14165. URL: https://arxiv.org/abs/2005.14165.

[3] Sumanth Dathathri et al. "Plug and Play Language Models: A Simple Approach to Controlled Text Generation". In: *CoRR* abs/1912.02164 (2019). arXiv: 1912.02164. URL: http://arxiv.org/abs/1912.02164.

[4] Tianyu Gao, Xingcheng Yao, and Danqi Chen. "Simcse: Simple contrastive learning of sentence embeddings". In: *arXiv preprint arXiv:2104.08821* (2021).

[5] Long Ouyang et al. *Training language models to follow instructions with human feedback*. 2022. DOI: 10.48550/ARXIV.2203.02155. URL: https://arxiv.org/abs/2203.02155.

[6] Alec Radford et al. "Language Models are Unsupervised Multitask Learners". In: (2019).

[7] Ross Taylor et al. *Galactica: A Large Language Model for Science*. 2022. DOI: 10.48550/ARXIV.2211.09085. URL: https://arxiv.org/abs/2211.09085.

[8] Jason Wei et al. *Chain of Thought Prompting Elicits Reasoning in Large Language Models*. 2022. DOI: 10.48550/ARXIV.2201.11903. URL: https://arxiv.org/abs/2201.11903.

[9] Kevin Yang and Dan Klein. "FUDGE: Controlled Text Generation With Future Discriminators". In: *ArXiv* abs/2104.05218 (2021).

[10] Yanek Yuk. *BERT Uncased Keyword Extractor*. 2016. URL: https://huggingface.co/yanekyuk/bert-uncased-keyword-extractor.

[11] Susan Zhang et al. *OPT: Open Pre-trained Transformer Language Models*. 2022. DOI: 10.48550/ARXIV.2205.01068. URL: https://arxiv.org/abs/2205.01068.