RetGen: A Joint framework for Retrieval and Grounded Text Generation Modeling

Yizhe Zhang * Siqi Sun Xiang Gao Yuwei Fang
Chris Brockett Michel Galley Jianfeng Gao Bill Dolan
Microsoft Corporation, Redmond, WA, USA
yizhezhang@fb.com, {siqi.sun,xiag,yuwfan,mgalley,chrisbkt,jfgao,billdol}@microsoft.com

Abstract

Recent advances in large-scale pre-training such as GPT-3 allow seemingly high quality text to be generated from a given prompt. However, such generation systems often suffer from problems of hallucinated facts, and are not inherently designed to incorporate useful external information. Grounded generation models appear to offer remedies, but their training typically relies on rarely-available parallel data where information-relevant documents are provided for context. We propose a framework that alleviates this data constraint by jointly training a grounded generator and document retriever on the language model signal. The model learns to reward retrieval of the documents with the highest utility in generation, and attentively combines them using a Mixture-of-Experts (MoE) ensemble to generate follow-on text. We demonstrate that both generator and retriever can take advantage of this joint training and work synergistically to produce more informative and relevant text in both prose and dialogue generation.

1 Introduction

Recent large-scale pre-trained language models (LMs) such as BERT (Devlin et al. 2019), GPT-2 (Radford et al. 2019), GPT-3 (Brown et al. 2020), and T5 (Raffel et al. 2019) have brought numerous breakthroughs in natural language generation (NLG) across a variety of tasks. These models, however, are not designed to leverage external information to enhance or to verify the predicted text. Gao et al. (2020), for example, demonstrate that they fail to reliably generate responses grounded in real-world knowledge, and may fall short when generating goal-directed responses that are optimized for information-seeking task completion. These models pose several challenges in information-demanding scenarios: First, they are usually trained offline, rendering the model agnostic to the latest information (*e.g.*, asking a chat-bot trained from 2011-2018 about COVID-19). Second, they are mostly trained on public data, rendering them less suitable in scenarios where customized or personalized information must be processed (*e.g.*, writing suggestions based on private user-data). Third, even in scenarios that call only for public information, generation from these LMs may be unfaithful to the facts (*e.g.*, hallucinations about birth dates), especially when the people or entities are less well known and the scenario demands a high degree of fidelity. As a practical matter, moreover, there remains a fundamental capacity issue in that large LMs cannot effectively represent all the information about every person or entity in the world.

A solution that would at first glance seem obvious is to ground the language model in real-world knowledge, which can be present in either structured data (*e.g.*, a knowledge-graph) or unstructured data (*e.g.*, documents such as Wikipedia, user documents or background stories) (Wu et al. 2021; Ghazvininejad et al. 2018; Dinan et al. 2019; Qin et al. 2019). The advantage of using unstructured grounding data over structured data is that the former provides richer information and it is typically more flexible when it comes to maintaining and updating the information base. However, training a grounded text generation model that takes additional unstructured documents as input typically demands that the training data contains pairs of context and corresponding oracle documents. These pairs are seldom available. Recent work, such as REALM (Guu et al. 2020) and RAG (Lewis et al. 2020b), attempts to leverage information retrieval machinery in real time to mitigate this data paucity in open-domain question answering systems. The approach taken in this paper is in similar vein, but is not confined to the specialized case of question answering, and seeks to present a mechanism to that addresses the broader problem of informational accuracy in text generation.

In this paper, we investigate the task of generating text by potentially taking advantage from massive reference documents. We called this task as *Information-aware Text Generation* (ITG). Note that ITG is related but different from open-domain Question Answering (ODQA). In ODQA, the input is typically an information query/question, and the generation is expected to be the answer for that query. In ITG, the input is usually not an information query/question. The task is to potentially leverage any possible external reference documents to predict the next utterance or sentence. Unlike the ODQA, ITG might not always directly seek an answer from retrieved documents, instead it usually requires the retrieved information to be digested as context to subtly influence the generation. Therefore, ITG can be applied to scenarios like dialogue generation and text auto-completion which generalize the open-domain QA scenario.

Below, we present a large-scale general purpose pre-

*Work done at Microsoft Research

training framework that jointly trains a document retriever and a multi-document grounded generator in end-to-end fashion and allows these to synergistically cooperate to optimize grounded text generation. Our method first selects and scores a collection of documents that are most helpful to generation according to the language model signal. The multi-document generator then digests these documents and combines their information according to document-specific attention weights to generate a single prediction in a Mixture-of-Experts (MoE) manner. The main contributions are as follows:

- We provide a *joint training framework* for grounded generation and document retrieval with a language model signal. Our method alleviates the need for oracle parallel data (prose-document pairs) with which to train a grounded model, enabling the use of massive non-parallel corpora.
- From the *retriever's perspective*, our approach uses the language model signal to optimize the retriever, so that the documents with highest utility in generation are returned.
- From the *generator's perspective*, our approach learns to attend to and combine multiple retrieved documents to achieve a mixture-of-expert-based generation. We apply mutual information maximization (MMI) to further enhance the model.

2 Related Work

Retrieval-Augmented Language Modeling A series of previous work explores a retrieve-then-edit paradigm for text generation (Peng et al. 2019; Li et al. 2018; Cai et al. 2019b; Hashimoto et al. 2018; Yang et al. 2019; Song et al. 2018; Cai et al. 2019a; Wu et al. 2019). This line of work either directly edits the retrieved text, or feeds the retrieved text to a fixed generator. REALM (Guu et al. 2020) has proposed a Retrieval-augmented encoder to extract salient text span for open-domain QA. The knowledge retriever is pre-trained by leveraging the masked language model signal. RAG (Lewis et al. 2020b) fine-tunes models that can leverage the Wikipedia documents to facilitate knowledge-intensive NLP tasks, and achieves strong performance on open-domain QA. Our approach differs in that: 1) we update the document encoder during training, whereas the document encoder in RAG is fixed. The optimizable document encoder enables us to investigate whether language model signals can be leveraged to improve the document representation learning. Regularly indexing millions of documents is also technically challenging. 2) we incorporate MMI and retriever correction to further improve the performance. 3) we focus on an information-aware text generation (ITG) task, which is related but different from open-domain QA. Lewis et al. (2020a) proposed a pre-training objective to reconstruct the original document from retrieved evidence documents, and employ the resulting model to improve translation and summarization results. The bulk of recent work has attempted to perform retrieval-augmented generation to either task-oriented (Thulke et al. 2021) or open-domain (Shuster et al. 2021) response generation. However, their retrievers are not optimized during the training, an thus may be unable to learn from the rich language model signals.

Dense Retrieval Models Unlike standard information retrieval techniques such as BM25, Dense Retrieval (DR) models map documents and queries into an embedding space and match them according to semantic similarity. Representative works include (Lee, Chang, and Toutanova 2019; Karpukhin et al. 2020; Luan et al. 2020; Xiong et al. 2020), which achieve the state-of-the-art performance in tasks like open-domain QA and relevance ranking. Such dense retrieval models can be fine-tuned to accommodate customized needs, and have become a core component in many natural language systems (Khandelwal et al. 2019; Guu et al. 2020).

Grounded Generation Grounded generation based on external knowledge has been extensively studied. Some previous work leverages *structured* external sources like relational knowledge bases (Zhu et al. 2017; Liu et al. 2018) or knowledge graphs (Young et al. 2018) for conversation generation. More recently, Liu et al. (2019) have developed a hybrid graph-path-based method on knowledge graphs augmented with unstructured grounding information. Our work focuses on unstructured (raw text) grounding information and thus avoids the need of pre-constructed knowledge graphs. Peng et al. (2020) grounds the task-oriented response generation on the retrieved database states.

Another line of research exclusively uses the *unstructured* grounding. Ghazvininejad et al. (2018) developed a memory network based model to leverage grounding information from Foursquare. Dinan et al. (2019) crowd-sourced conversations where each utterance is grounded in no more than a single sentence. Zhou, Prabhumoye, and Black (2018) collected a dataset for grounded conversation generation. Qin et al. (2019) employed a machine reading comprehension (MRC) model to extract salient grounding information to facilitate generation. Wu et al. (2021) used a controllable generation framework to generate dialogue responses by applying extracted lexical constraints. Zhao et al. (2020) equipped response generation with knowledge selection module. Annotated grounding in these works is often ad-hoc and not necessarily optimal for the task. Our work differs from these in that we jointly train a retriever and generator to optimize grounded text generation performance, and our proposed model does not rely on annotated text-reference parallel data, with the result that it can be trained on any target dataset without additional annotation.

3 Method

Method Overview

We begin by formally defining our *Information-aware Text Generation* (ITG) task and laying out necessary notation. ITG aims to predict the upcoming text y that directly follows the existing source prompt x (x, y are from a corpus $\mathbf{D}$), while a document reference set $\mathbf{Z}$ is accessible. In ITG, $\mathbf{D}$ and $\mathbf{Z}$ are *non-parallel* to each other. In other words, each x is paired with a y. However, the association between a document z in $\mathbf{Z}$ and the tuple (x, y) is not necessarily known.

We propose a framework called *RetGen* to solve the ITG task. RetGen has two components: $i)$ a dense document retriever and $ii)$ a knowledge-grounded text generator. The objective of the ITG is to train a model to maximize the

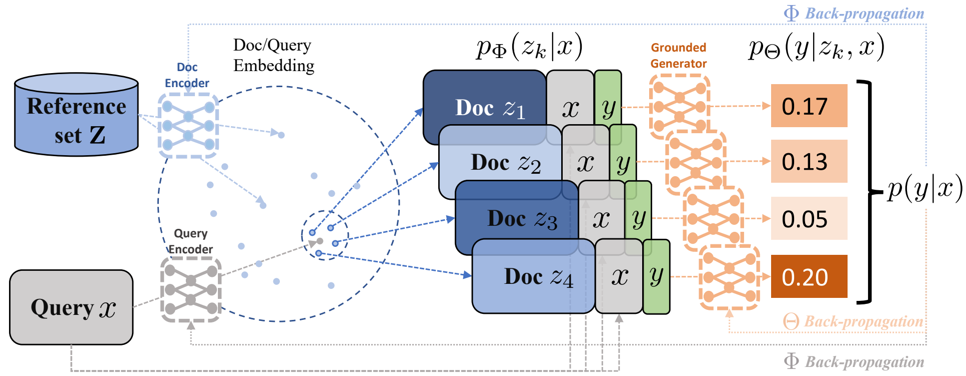


Figure 1: An overview of the RetGen framework. A source context query x and documents from a reference database $\mathbf{Z}$ are first mapped to a joint embedding space via different encoders. A Maximum Inner Product Search is performed to retrieve top-K relevant documents ($K = 4$ in this figure) with their probability score $p(z_k|x)$. The retrieved documents are separately concatenated with query x and target upcoming text y and passed through a grounded text generator, to compute the document-dependent likelihood $p(y|z_k, x)$. The final objective $p(y|x)$ given by (2) is optimized to update the retriever parameters Φ and generator parameters Θ.

likelihood of y given x and $\mathbf{Z}$. Formally, it optimizes

$$p(y|x; \mathbf{Z}) = \sum_{z \in \mathbf{Z}} p(y|x, z)p(z|x), \tag{1}$$

In practice, $\mathbf{Z}$ often contains millions of documents, rendering enumeration over z impossible. Instead, we leverage a *dense document retriever* $r_\Phi(\cdot)$ to dramatically narrow down the search to a handful relevant documents, where Φ denotes the retriever parameters. r_Φ takes $\mathbf{Z}$ and x as input and yields relevance scores $\{s_1, \cdots, s_K\}$ of the top-K (K is a hyper-parameter) documents $\tilde{\mathbf{Z}} = \{z^{(1)}, \cdots, z^{(K)}\}$.

We further denote the *knowledge-grounded text generator* as $g_\Theta(\cdot)$, where Θ denotes the generator parameters. This generator module uses x and a single document z as input to produce a probability score for a given reference target y, *i.e.*, $g_\Theta(y|x, z) = p(y|x, z)$. The loss can be approximated as:

$$\mathcal{L}(\Theta, \Phi) = -\log \sum_{k=1}^{K} p_\Theta(y|x, z^{(k)}) p_\Phi(z^{(k)}|x), \tag{2}$$

where $p(z^{(k)}|x) = \exp(s_k)/\sum_{i=1}^{K} \exp(s_i)$ is the normalized probability (with relevance scores s as logits), and $\tilde{\mathbf{Z}} = \{z^{(1)}, \cdots, z^{(K)}\}$ are retrieved from $r_\Phi(\mathbf{Z}, x)$. An overview of the model is presented in Figure 1.

Document Retriever

For the dense document retriever corresponding to $p_\Phi(\cdot)$ in (2), we leverage a model similar to that of (Karpukhin et al. 2020; Xiong et al. 2020) to achieve efficient document retrieval with sub-linear time. The documents Z and context queries x are mapped into the same dense embedding space. The relevance score $s(x, z)$ is computed as the vector inner product between document embedding $h_z = f_z(z)$ and query embedding $h_x = f_x(x)$, *i.e.*, $s(x, z) = h_x^T h_z$, where $f_z(\cdot)$ and $f_x(\cdot)$ represent learnable encoding networks for document and query respectively. $p(z^{(k)}|x)$ in (2) is finally given by *softmax*$^{(k)}(s(x, \tilde{\mathbf{Z}}))$.

To achieve sub-linear searching time, Maximum Inner Product Search (MIPS) (Shrivastava and Li 2014) is employed. The document embedding vectors are pre-computed and indexed using Locality Sensitivity Hashing (LSH) (Datar et al. 2004), so that the query vector can be hashed to a cluster of relatively relevant documents. This search strategy is approximate. However it yields good empirical search results when the document set is large. In practice, we use ANCE (Xiong et al. 2020) to initialize the retriever.

Knowledge-Grounded Text Generator

For the knowledge-grounded text generator (GTG) corresponding to $p_\Theta(\cdot)$ in (2), we employ a transformer-based architecture akin to GPT-2 (Radford et al. 2019). The GTG takes one document z and one context query x as the input, and the following text y as the target reference. Specifically, the z and x are first concatenated by a special separator token. The training objective follows a standard language model (LM) loss (Radford et al. 2019; Zhang et al. 2020):

$$p_\Theta(y|x, z) = \prod_{t=0}^{|y|} p(y_t|x, z, y_{0:t-1}), \tag{3}$$

where y_t is the t-th token in y. $y_{i:j}$ denotes $\{y_i, \cdots, y_j\}$ and $|\cdot|$ denotes the cardinality. As opposed to GPT-2, we assign different token type embeddings to the tokens in z and x to help the model identify document and context.

We also employ a distinctive design for the position embedding. The document position id starts with M ($M = 400$ in our experiment [1]) while the context position id starts with 0. The intent is to maximally separate z and x, thus reducing the chance that the model will be exposed to hints that z is part of the preceding context.[2] We found this facilitates the

[1] we only select instances with context/response tokens $\leq 256/128$. Thus the maximum number of tokens is around 400.

[2] GTGs are initialized from GPT-2/DialoGPT, which were trained to recognize tokens with continuous position id as coherent text.

model in differentiating the document and the context, and in applying different strategies specific to each.

Joint Training

During the training time, Θ can be directly optimized from (2). We optimize the objective in (2) with respect to Φ by an estimation resembles the Actor-Critic (AC),

$$\begin{aligned}\nabla_\Phi p(y|x) &= \sum_z p(y|z,x)\nabla_\Phi p(z|x) \\ &= \sum_z p(y|z,x)p(z|x)\nabla_\Phi \log p(z|x) \end{aligned} \quad (4)$$

where the C is a constant baseline. The last step is because $\sum_z \nabla_\Phi p(z|x) \log p(z|x) = \nabla_\Phi \sum_z p(z|x) = 0$. C is commonly referred as a "control variate" (Williams 1992; Nelson 1990) and used to reduce the variance in Monte Carlo estimation as in (4). The $p(y|z,x)$ can be viewed as the "return" in the Actor-Critic algorithm. Document z will receive a positive update if it yields $p(y|z,x)$ larger than to the average performance of the retrieved documents. In our experiment, we set C as the expected reward, *i.e.*, $C = \sum_{z \in \tilde{\mathbf{Z}}} p(y|z,x)p(z|x)$. Finetuning the retriever model based on (4) needs good initializations from pretrained models to avoid cold-starting.

Another practical challenge is that all the document embedding vectors need to be *refreshed* once the retriever is updated, which is expensive when the number of documents is large. Instead of encoding all the documents each time the Φ is updated to retrieve the top-K document set $\tilde{\mathbf{Z}}$, we asynchronously update the retrieved document for every M steps ($M = 200$ in our experiments). However, note that even if the $\tilde{\mathbf{Z}}$ is fixed for each K steps, the r_Φ and scores $\{s_1, \cdots, s_K\}$ are still updated at every step.

Multi-document Decoding

Mixture-of-Expert (MoE) Decoder During the inference time, the retriever first obtains the top-K documents as $\tilde{\mathbf{Z}}$, and their corresponding probabilities $p(z|x)$. The generator leverages all document in $\tilde{\mathbf{Z}}$ to generator a consensus prediction $\hat{y}$. One naive approach is to concatenate multiple documents into a "joint" document as the input for the generator. The problems for such an approach are that 1) the "joint" document may be too long to be efficiently processed; 2) the order of the documents has impact on the generation; 3) the relevance information $p(z|x)$ will be ignored.

We therefore took a Mixture-of-Expert (MoE) approach following Cho et al. (2020) to decode the model in a document-specific fashion, and ensemble the output distributions at each time step. Specifically, we leverage K copies of the ground text generator $g_\Theta(\cdot)$ trained from (2). At time step t , we feed each copy of the generator with separate document z, the same context x, and the same current consensus generation $\hat{y}_{0:t-1}$. We then harvest the individual output distribution from all generator copies, as $\{p(\hat{y}_t^{(1)}), \cdots, p(\hat{y}_t^{(K)})\}$. The assembled output distribution at step t is finally given by

$$p(\hat{y}_t|x, \hat{y}_{0:t-1}) = \sum_{k=1}^{K} p(\hat{y}_t^{(k)}|z^{(k)}, x, \hat{y}_{0:t-1})p(z^{(k)}|x). \quad (5)$$

Unlike recent FiD work (Izacard and Grave 2020), which "fuses" the encoded representations from different documents, our "fusion" of information from different document occurs at output token distribution level. FiD requires training a grounded generation model by taking a fixed number of documents as input. However, our MoE approach can directly leverage a grounded generation model trained on single document as input, without additional training or fine-tuning. This yields convenience and flexibility in the number of documents K to leverage for inference.

We also employ a novel *retriever correction* on the decoder to accommodate the fact that the model is trained to *autoregressively* generate y, which implies that the retriever score needs to be updated along the generation. Details are provided in Appendix B.

MMI We further implement a Maximum Mutual Information (MMI) scoring function (Li et al. 2016; Zhang et al. 2018) to enhance the "groundness" of the generation. MMI employs a pre-trained *backward* grounded text generation model to predict x and z from given prediction y, *i.e.*, $p(x, z|y)$.[3] We first generate a set of hypotheses using top-K sampling. Then we use the probability of $p(x, z|y)$. For multiple z we use the mean of these probabilities to rerank all hypotheses. Intuitively, maximizing backward model likelihood penalizes the bland hypotheses (Zhang et al. 2020) and encourages the generation y to tie better to the input document z and context x.

4 Experimental Setups

Datasets We use two datasets $\mathbf{D}$, Reddit and arXiv, which cover two information-demanding scenarios (response generation and prose generation) , to evaluate our methods.

The ***Reddit*** dataset contains 2.56M/2K/2K training/validation/test conversation instances. The training set is created using the extraction pipeline from the DSTC-7 grounded response generation challenge (Galley et al. 2019), which extracts the Reddit threads with time span 2011-2017. It contains threads of discussions like a tree, where a reply is a child node to the previous message. Any path along the tree introduces a dialogue session. Although our approach does not require parallel oracle document, we only select instances that oracle document [4] can be found. the reasons are two-fold: 1) we hope to be able to build strong baselines grounded on oracle dataset, which characterize the upper bound performance of a grounded generation model; 2) the oracle documents enable separately evaluating the retriever performance using IR metrics (Recall@K). We further select test examples from Reddit with time span 2018-2019 by requiring the context to have at least 6 different responses. This yields a 5-reference test set with 2,000 samples. For each instance, one of the 6 human responses is set aside to assess human performance. The average length of the context and response is 44.32 and 14.86 words, respectively.

The ***arXiv*** dataset is based on Clement et al. (2019) , which collects a corpus of arXiv articles from 1991 to 2019. We

[3] This objective is designed to encourage y to incorporate information from both x and z.

[4] containing URLs linking to Wikipedia domain, see Appendix C

construct the context and target pairs using the abstracts. The final resulting train/validation/test contains 9.6M/57K/2K instances from 1.67M unique abstracts. No parallel oracle document is available for this dataset.

For the ***reference*** dataset $\mathbf{Z}$, we extract about 5.7 million documents from Wikipedia dump of December 2018. For each entry, we only extract the first two paragraphs as these are typically most relevant and summarize the entire document. In addition, we truncate overlong sentences to 100 words, and remove the entry if it contains only one sentence.

More dataset details are provided in Appendix C.

Evaluation Metrics We performed automatic evaluation using standard machine translation metrics, including BLEU (Papineni et al. 2002), METEOR (Lavie and Agarwal 2007), and NIST (Doddington 2002). NIST is a variant of BLEU that weights n-gram matches by their information gain, i.e., it indirectly penalizes uninformative n-grams. Following Zhang et al. (2020), we also use Entropy (Zhang et al. 2018) and Dist-n (Li et al. 2016) to evaluate lexical diversity. For the Reddit dataset, where 5 references are available for each instance, we compute all relevance metrics and aggregate all of them using max-pooling.

To evaluate how well the predicted text $\hat{y}$ reflects the external information, we propose an evaluation score which we call a *Keyword Matching Ratio (KMR)*. KMR is defined as

$$\text{K-words} = \text{set}(z) \setminus \text{set}(x),$$
$$\text{KMR} = |\text{set}(\hat{y}) \cap \text{K-words}|/|\text{K-words}|,$$

where $\cap, \setminus, |\cdot|$ denotes the set intersection, difference and cardinality, respectively. For each bag-of-word sets (*i.e.*, $\text{set}(\hat{y}), \text{set}(z), \text{set}(x)$), stop words based on the python NLTK module and frequency in the corpus are removed. Intuitively, K-words reflect important information (a set of keywords) in the reference documents z but not covered by context x. KMR calculates the percentage of these keywords covered by the predicted text y. Such a metric assesses the utilization of external information but not the relevance. If a model generates reasonable follow-up text, but fails to incorporate important external information in z, KMR will still be low.

Baselines & Model setups We compared RetGen with several baselines in both datasets. The **DialoGPT**(345M) (Zhang et al. 2020) and **GPT-2**(345M) baselines are obtained by fine-tuning the original pre-trained models on the target training dataset to alleviate the dataset-shifting bias. For the Reddit dataset, since we have the oracle document for each conversation, it is possible to train a ground generation model as described in section §3 by directly grounding on the oracle documents. This model is denoted as *gDPT* (grounded DialoGPT). **gDPT (w/ oracle doc)** and **gDPT (w/ random doc)** denote the generation from gDPT model (described in §4) using oracle and random document, respectively. These two models set up the upper and lower bounds of the performance of the grounded generator g_Θ.

For each dataset, we evaluate 4 variants of RetGen: i) **RetGen (K=1)** uses only top-1 retrieved document to generate text; ii) **RetGen (K=4)** uses all top-4 retrieved documents for generation; iii) **Fixed** Φ is an ablation of RetGen (K=4) where the retriever parameters Φ are frozen during the training; iv) **+MMI** is a variant of RetGen (K=4) using MMI, (§3) (Li et al. 2016; Zhang et al. 2018). We first generate 16 hypotheses using top-10 sampling, then select the top hypothesis using reverse model probability of $p(z, x|y)$. The reverse model is also a 345M model fine-tuned from DialoGPT/GPT-2 using the same dataset.

Note that we only perform fine-tuning on existing pre-trained LMs and dense retrievers. All the grounded generators use the same transformer architectures and are initialized with DialoGPT/GPT-2 (345M) weights. The dense retrievers are initialized from ANCE (Xiong et al. 2020). For the retriever training, we index the documents for each 200 iterations. Models are trained on workstations with 8 Nvidia V100 GPUs. During training, we use $K = 4$ for RetGen.

More model setup details are provided in Appendix D.

5 Results

Generation Evaluation The automatic evaluation results are summarized in Table 1 (the standard deviation of the results are provided in the Appendix F). We observe that freezing the retriever to pretrained ANCE yield suboptimal evaluation metrics by comparing **Fixed** Φ and **RetGen (K=4)**. This implies that retriever fine-tuning is crucial to adapt the retriever to the generation task. Consistent with the observations in Zhang et al. (2020), the **MMI** re-ranking procedure produces more diverse text and achieves higher NIST and METEOR scores, albeit with a slight drop in BLEU. We presume the inconsistency is because NIST generally rewards more for informative and low-frequency n-grams. Incorporating additional information from retrieved documents presumably makes the generation to be more informative diverse. On the Reddit dataset, **RetGen (K=4)** achieves comparable performance to **RetGen (w/ oracle doc)**, indicating the retrieved documents are of high quality.

We also compute KMR, which evaluates the utility of the external document z for generating text y, as described in §4. For the Reddit dataset, the KMR for gDPT and the human oracle[5] is calculated against oracle document. Otherwise, KMR is calculated against the retrieved documents by performing a max-pooling over document-specific KMR. As expected, RetGen with MMI generally achieves the highest KMR, as it explicitly maximizes the mutual information between the documents and the output. For both datasets, RetGen with more documents and with trainable retriever achieves a higher KMR. Note that KMR may not necessarily be associated with generation quality. However, except for MMI, a higher KMR indicates the model is more effective in leveraging the external document to optimize the LM objectives.

Note that for some metrics the systems achieve higher score than human oracle. As discussed in Zhang et al. (2020), this observation does not imply that the machine generation achieves human parity, but is presumably an artifact of the randomness of human responses in the data.

Generated examples We provide generated examples for both datasets in Table 2. The RetGen examples are from our

[5]The human oracle only provides a reference baseline and may not be comparable with the compared systems.

Method	NIST N-2	NIST N-4	BLEU B-2	BLEU B-4	MET-EOR	Entropy E-4	Dist D-1	Dist D-2	Avg. Len.	KMR
***Reddit* dataset**										
DialoGPT	1.59	1.60	12.41%	2.34%	7.23%	8.34	13.2%	32.8%	12.0	-
gDPT (w/ oracle doc)	2.37	2.39	12.58%	2.57%	7.41%	9.04	13.0%	33.2%	15.1	4.8%
gDPT (w/ random doc)	2.03	2.05	10.14%	1.91%	7.12%	9.03	9.9%	27.2%	18.0	2.8%
RetGen ($K = 1$)	2.39	2.41	12.29%	2.32%	7.43%	9.33	14.1%	37.6%	15.6	4.9%
RetGen ($K = 4$)	2.40	2.42	**12.53%**	**2.52%**	7.47%	9.36	14.5%	38.7%	15.3	5.2%
RetGen ($K = 4$, Fixed Φ)	2.37	2.39	11.72%	2.31%	7.63%	9.21	12.9%	34.6%	16.9	4.3%
RetGen ($K = 4$, +MMI)	**2.44**	**2.46**	10.98%	1.70%	**8.04%**	**10.30**	**18.6%**	**60.0%**	18.5	**6.3%**
Human oracle	2.13	2.15	13.39%	4.25%	7.34%	9.89	28.2%	77.1%	12.9	5.9%
***arXiv* dataset**										
GPT-2	1.04	1.07	9.85%	3.81%	8.59%	9.34	20.7%	51.3%	18.6	-
RetGen ($K = 1$)	1.81	1.84	11.75%	4.19%	9.04%	9.58	17.5%	46.1%	23.6	3.7%
RetGen ($K = 4$)	**1.82**	**1.86**	**11.85%**	**4.35%**	**9.04%**	9.57	17.5%	46.0%	23.7	3.8%
RetGen ($K = 4$, Fixed Φ)	1.78	1.81	11.79%	4.32%	9.01%	9.56	17.6%	46.4%	23.4	3.7%
RetGen ($K = 4$, +MMI)	1.81	1.84	10.84%	3.32%	8.73%	**10.06**	19.2%	**59.0%**	28.2	**4.0%**
Human oracle	-	-	-	-	-	9.95	24.7%	71.7%	24.4	-

Table 1: Automatic evaluation results on the Reddit (upper) and arXiv (lower) datasets. gDPT w/ oracle(random) doc denotes a grounded generation model directly using oracle(random) document. Fixed Φ denotes only fine-tuning generator parameters Θ while freezing the initial retriever parameters Φ in ANCE. +MMI represents post-ranking with MMI.

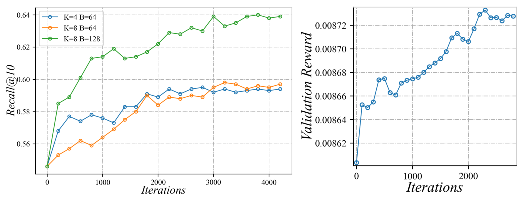
	Reddit dataset	ArXiv dataset
Context	TIL: All arcade games imported into North America from 1989 to 2000 had the following FBI slogan included into their attract mode: **Winners Don't Use Drugs**.	(Title: It from Knot) **Knot physics** is the theory of the universe that not only unified all the fundamental interactions but also explores the underlying physics of quantum mechanics.
(X)PT	I'm pretty sure that's the slogan of the game in question.	The theory of the knot is a new approach to quantum mechanics.
RetGen	I have a feeling a major part of the reason was **Nixon** was in charge during that period of history.	A knot is a **finite** sequence of **discrete quantum states** that represent the **gravitational field** in a quantum theory.
Retrieved docu-ment(s)	**Winners Don't Use Drugs** is an anti-drug slogan that was included in arcade games ... **The slogan was part of a long-term effort** by the United States in its war on drugs started by President Richard **Nixon** in 1971 *(Winners Don't Use Drugs)*	In loop quantum gravity, ... , s-knots represent **the quantum states of the gravitational field** *(S-knot)* Knots have been used for ... Modern physics demonstrates that the **discrete** wavelengths depend on quantum energy levels. ... the Jones polynomial and its generalizations, called the **finite** type invariants... *(History of knot theory)*

Table 2: Generated examples for Reddit (left) and arXiv (right). (X)PT denotes DialoGPT/GPT. The relevant parts are **highlighted**, and the title of the most relevant retrieved Wikipedia entries are shown in *(article title)*.

best system. In general, RetGen demonstrates ability to integrate information from difference sources including context and multiple references, and sometimes generate text that reflects multi-hop cross-reference among all the sources. We empirically observe that the retrieved documents are usually relevant and may cover orthogonal aspects of the topics in the context. We also visualize how the document attention weights $p(z|x, y_{0:t-1})$ change during the generation process in Appendix H. We observed that the attention distribution over documents generally becomes more peaked over steps of generation, indicating the model become more certain as generation proceeds.

Nevertheless, we observe several *failure modes* in our experiments with RetGen: $i)$ the retrieved passage may not always be relevant and correct. We find that the RetGen can learn to be inclined to avoid using irrelevant documents, but we still see cases where poorly retrieved documents result in incorporation of hallucinations or irrelevant information in final generation; $ii)$ the retrieved passage is relevant, but the grounded generation model may miss correct information and incorporate similar but incorrect/irrelevant information in the generated text (*e.g.*, when asked about who Barack Obama's grandfather was, the system offers his father's name which is also in the retrieved document). These issues do not dominate in our experiments, but resolving them is important and warrants further investigation. We provide examples of above problems in Appendix G.

Impact of number of documents K From Table 1 , RetGen with $K = 4$ consistently obtains higher NIST, BLEU and METEOR scores compared with $K = 1$, indicating that incorporating multiple retrieved documents may provide better coverage of the references. We also evaluate RetGen with K ranging from 1 to 4 (Appendix E), demonstrating monotonic improvement when K increases. We observe no significant improvement with $K > 4$.



(a) Recall@10 (Reddit) (b) Reward (arXiv)

Figure 2: Recall/Reward on validation set can improve during retriever-only training.

Reddit					arXiv				
System A		Neutral	System B		System A		Neutral	System B	
Coherence: *A and B, which is more relevant to, and coherent with the context?*									
RetGen	**43.7**%	28.3%	28.0%	DialoGPT *	RetGen	**32.1**%	41.7%	26.3%	GPT-2
RetGen	33.3%	28.6%	**38.1**%	MMI	RetGen	29.9%	38.7%	**31.5**%	MMI
RetGen	**40.9**%	22.9%	36.3%	Human *	RetGen	**34.9**%	35.2%	29.9%	Human *
MMI	**45.9**%	23.1%	31.0%	Human *	MMI	**34.9**%	35.8%	29.3%	Human
Informativeness: *A and B, which is more informative (usually more specific content)?*									
RetGen	**44.5**%	27.8%	27.7%	DialoGPT	RetGen	**36.3**%	37.2%	26.5%	GPT-2
RetGen	32.7%	28.3%	**39.0**%	MMI	RetGen	28.9%	37.9%	**33.2**%	MMI
RetGen	**41.1**%	21.5%	37.5%	Human	RetGen	33.2%	32.4%	**34.4**%	Human
MMI	**47.5**%	21.1%	31.4%	Human *	MMI	**34.2**%	34.7%	31.1%	Human
Human-likeness: *A and B, which is more likely to be generated by human rather than a machine?*									
RetGen	**36.4**%	34.0%	29.6%	DialoGPT	RetGen	**29.7**%	43.6%	26.7%	GPT-2
RetGen	31.3%	33.9%	**34.9**%	MMI	RetGen	**28.6**%	42.9%	28.5%	MMI
RetGen	**40.1**%	28.5%	31.4%	Human *	RetGen	**33.7**%	38.9%	27.5%	Human *
MMI	**40.5**%	28.3%	31.1%	Human *	MMI	**33.1**%	38.3%	28.7%	Human

Table 3: Results of **Human Evaluation** for coherence, informativeness and human-text possibility, showing preferences (%) for our model (RetGen) vis-a-vis baselines and real human responses. **RetGen** denotes RetGen with $K = 4$, and **MMI** represents RetGen with MMI. Numbers in bold indicate the preferred systems. Statistically significant results with p-value ≤ 0.05 are indicated by *.

Retrieval Evaluation From Table 1, it can be seen that optimizing retriever leads to better automatic metrics compared with generator-only training, indicating that the retriever can benefit from language model signal. However, evaluating the retriever improvement using generation metrics as in Table 1 is implicit, as the retriever evaluation and generation evaluation are coupled together. To explicitly assess how the retriever can benefit from joint training, we freeze the generator parameters Θ and only finetune the retriever parameters Φ, and monitor the training process of retriever using either ranking metrics or expected reward in (4).

For the Reddit dataset, since oracle documents are available, we monitor the progress of recall@10 during this retriever-only training. The recall value is computed by averaging over 2,000 validation examples. The total number of passage candidates is 10k, including 2k oracle documents for each instance, and 8k hard-negative documents that are close to these 2k oracle documents according to BM25. The results are provided in Figure 2 (a). With fluctuation, the recall generally improves as training progresses. Increasing the number of documents K from 4 to 8 brought only marginal gain in recall. However, increasing the number of examples in each batch led to more significant improvement of the recall.

For the arXiv dataset, since recall cannot be computed, we instead monitor the expected reward/return ($r = \sum_z p(y|z,x)p(z|x)$) over 2,000 validation examples. Our reasoning here is that with fixed Θ, if the reward can be improved (*i.e.*, the target y is more likely given the current z), the only possible explanation is that the retrieved documents are more relevant and helpful in predicting the oracle target. We observed that this reward metric can to some extent improve as training progresses (Figure 2 (b)). This verifies that the retriever is being optimized and benefits from LM signals.

Human Evaluation Overall judge preferences in each of the 3 categories are shown in Table 3, where the 5-point scale has been collapsed to a 3-point preference for clarity. A moderate preference can be observed for the variant of RetGen with MMI over vanilla RetGen. Table 3 suggests that the RetGen may begin to approximate human quality. As has been observed elsewhere, e.g., Zhang et al. (2020), we found that judges often prefer model generation over human responses. In the case of the Reddit dataset, we speculate that the original human responses may be more erratic and idiosyncratic than system outputs. Human evaluation of the arXiv dataset, meanwhile, is intrinsically difficult as responses typically involve domain knowledge: human judges may prefer system generated text that is potentially easier to understand.[6] How to evaluate generated text as systems improve remains a challenge, but further exploration of these issues falls beyond the scope of this work. Further details, including the human evaluation template used, are provided in the Appendix I.

6 Conclusion

We present a joint training framework to simultaneously optimize a dense passage retriever and a knowledge-grounded text generator in an end-to-end fashion. This approach enables leveraging the LM signal to optimize the information retrieval sub-component and thus permits the generation pipeline to output more informative text. The resulting algorithm leverages multiple retrieved documents during decoding time and generates text by selectively summarizing and combining information collected from all the references. We have demonstrated the effectiveness of this algorithm via crowd-sourced human evaluation and automatic evaluation that uses generation and retrieval metrics. In future work, we plan also to leverage QA and cloze task objectives for factuality evaluation (Eyal, Baumel, and Elhadad 2019; Huang, Wu, and Wang 2020). We discuss the ethical impact of this work in Appendix A.

[6]This is consistent with the findings in Freitag et al. (2021) for MT to the effect that crowd-sourced human evaluation is error-prone and may not be as accurate as some automatic metrics.

References

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; and et al., P. D. 2020. Language Models are Few-Shot Learners. *arXiv*.

Cai, D.; Wang, Y.; Bi, W.; Tu, Z.; Liu, X.; Lam, W.; and Shi, S. 2019a. Skeleton-to-Response: Dialogue Generation Guided by Retrieval Memory. In *NAACL*, 1219–1228.

Cai, D.; Wang, Y.; Bi, W.; Tu, Z.; Liu, X.; and Shi, S. 2019b. Retrieval-guided Dialogue Response Generation via a Matching-to-Generation Framework. In *EMNLP*.

Cho, W. S.; Zhang, Y.; Rao, S.; Celikyilmaz, A.; Xiong, C.; Gao, J.; Wang, M.; and Dolan, B. 2020. Contrastive Multi-document Question Generation. In *EACL*.

Clement, C. B.; Bierbaum, M.; O'Keeffe, K. P.; and Alemi, A. A. 2019. On the Use of ArXiv as a Dataset. *arXiv*.

Datar, M.; Immorlica, N.; Indyk, P.; and Mirrokni, V. S. 2004. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, 253–262.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Dinan, E.; Roller, S.; Shuster, K.; Fan, A.; Auli, M.; and Weston, J. 2019. Wizard of Wikipedia: Knowledge-Powered Conversational agents. In *ICLR*.

Doddington, G. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *ICHLTR*.

Eyal, M.; Baumel, T.; and Elhadad, M. 2019. Question Answering as an Automatic Evaluation Metric for News Article Summarization. In *NAACL*. Minneapolis, Minnesota.

Ferragina, P.; and Scaiella, U. 2010. TAGME: On-the-Fly Annotation of Short Text Fragments (by Wikipedia Entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, 1625–1628. New York, NY, USA: Association for Computing Machinery.

Freitag, M.; Foster, G.; Grangier, D.; Ratnakar, V.; Tan, Q.; and Macherey, W. 2021. Experts, Errors, and Context: A Large-Scale Study of Human Evaluation for Machine Translation. *arXiv*.

Galley, M.; Brockett, C.; Gao, X.; Gao, J.; and Dolan, B. 2019. Grounded response generation task at DSTC7. In *AAAI Dialog System Technology Challenges Workshop*.

Gao, J.; Peng, B.; Li, C.; Li, J.; Shayandeh, S.; Liden, L.; and Shum, H.-Y. 2020. Robust conversational AI with grounded text generation. *arXiv*.

Ghazvininejad, M.; Brockett, C.; Chang, M.-W.; Dolan, B.; Gao, J.; tau Yih, W.; and Galley, M. 2018. A Knowledge-Grounded Neural Conversation Model. In *AAAI*.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M.-W. 2020. REALM: Retrieval-augmented language model pre-training. *arXiv*.

Hashimoto, T. B.; Guu, K.; Oren, Y.; and Liang, P. 2018. A Retrieve-and-Edit Framework for Predicting Structured Outputs. In *NeurIPS*.

Huang, L.; Wu, L.; and Wang, L. 2020. Knowledge Graph-Augmented Abstractive Summarization with Semantic-Driven Cloze Reward. In *ACL*. Online.

Izacard, G.; and Grave, E. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv*.

Karpukhin, V.; Oğuz, B.; Min, S.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. *arXiv*.

Khandelwal, U.; Levy, O.; Jurafsky, D.; Zettlemoyer, L.; and Lewis, M. 2019. Generalization through Memorization: Nearest Neighbor Language Models. In *ICLR*.

Lavie, A.; and Agarwal, A. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, 228–231. Association for Computational Linguistics.

Lee, K.; Chang, M.-W.; and Toutanova, K. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *ACL*.

Lewis, M.; Ghazvininejad, M.; Ghosh, G.; Aghajanyan, A.; Wang, S.; and Zettlemoyer, L. 2020a. Pre-training via paraphrasing. *NeurIPS*.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. *arXiv*.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016. A diversity-promoting objective function for neural conversation models. *NAACL*.

Li, J.; Jia, R.; He, H.; and Liang, P. 2018. Delete, Retrieve, Generate: A Simple Approach to Sentiment and Style Transfer. In *NAACL*.

Liu, S.; Chen, H.; Ren, Z.; Feng, Y.; Liu, Q.; and Yin, D. 2018. Knowledge Diffusion for Neural Dialogue Generation. In *ACL*.

Liu, Z.; Niu, Z.-Y.; Wu, H.; and Wang, H. 2019. Knowledge Aware Conversation Generation with Explainable Reasoning over Augmented Graphs. In *EMNLP*.

Luan, Y.; Eisenstein, J.; Toutanova, K.; and Collins, M. 2020. Sparse, Dense, and Attentional Representations for Text Retrieval. *arXiv*.

Nelson, B. L. 1990. Control variate remedies. *Operations Research*, 38(6): 974–992.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. *ACL*.

Peng, B.; Li, C.; Li, J.; Shayandeh, S.; Liden, L.; and Gao, J. 2020. SOLOIST: Few-shot Task-Oriented Dialog with A Single Pre-trained Auto-regressive Model. *arXiv*.

Peng, H.; Parikh, A. P.; Faruqui, M.; Dhingra, B.; and Das, D. 2019. Text Generation with Exemplar-based Adaptive Decoding. In *NAACL*.

Qin, L.; Galley, M.; Brockett, C.; Liu, X.; Gao, X.; Dolan, B.; Choi, Y.; and Gao, J. 2019. Conversing by Reading:

Contentful Neural Conversation with On-demand Machine Reading. In *ACL*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv*.

Shrivastava, A.; and Li, P. 2014. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). *NeurIPS*.

Shuster, K.; Poff, S.; Chen, M.; Kiela, D.; and Weston, J. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. *arXiv*.

Song, Y.; Li, C.-T.; Nie, J.-Y.; Zhang, M.; Zhao, D.; and Yan, R. 2018. An ensemble of retrieval-based and generation-based human-computer conversation systems. In *IJCAI*, 4382–4388.

Thulke, D.; Daheim, N.; Dugast, C.; and Ney, H. 2021. Efficient Retrieval Augmented Generation from Unstructured Knowledge for Task-Oriented Dialog. *arXiv*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4): 229–256.

Wu, Y.; Wei, F.; Huang, S.; Wang, Y.; Li, Z.; and Zhou, M. 2019. Response generation by context-aware prototype editing. In *AAAI*, 7281–7288.

Wu, Z.; Galley, M.; Brockett, C.; Zhang, Y.; Gao, X.; Quirk, C.; Koncel-Kedziorski, R.; Gao, J.; Hajishirzi, H.; Ostendorf, M.; and Dolan, B. 2021. A Controllable Model of Grounded Response Generation. In *AAAI*.

Xiong, L.; Xiong, C.; Li, Y.; Tang, K.-F.; Liu, J.; Bennett, P.; Ahmed, J.; and Overwijk, A. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv*.

Yang, L.; Hu, J.; Qiu, M.; Qu, C.; Gao, J.; Croft, W. B.; Liu, X.; Shen, Y.; and Liu, J. 2019. A Hybrid Retrieval-Generation Neural Conversation Model. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1341–1350. ACM.

Young, T.; Cambria, E.; Chaturvedi, I.; Huang, M.; Zhou, H.; and Biswas, S. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. In *AAAI*.

Zhang, Y.; Galley, M.; Gao, J.; Gan, Z.; Li, X.; Brockett, C.; and Dolan, B. 2018. Generating informative and diverse conversational responses via adversarial information maximization. *NeurIPS*.

Zhang, Y.; Sun, S.; Galley, M.; Chen, Y.-C.; Brockett, C.; Gao, X.; Gao, J.; Liu, J.; and Dolan, B. 2020. DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. In *ACL, system demonstration*.

Zhao, X.; Wu, W.; Xu, C.; Tao, C.; Zhao, D.; and Yan, R. 2020. Knowledge-grounded dialogue generation with pre-trained language models. *arXiv preprint arXiv:2010.08824*.

Zhou, K.; Prabhumoye, S.; and Black, A. W. 2018. A Dataset for Document Grounded Conversations. In *EMNLP*, 708–713.

Zhu, W.; Mo, K.; Zhang, Y.; Zhu, Z.; Peng, X.; and Yang, Q. 2017. Flexible End-to-End Dialogue System for Knowledge Grounded Conversation. *arXiv*.