LLMR: Knowledge Distillation with a Large Language Model-Induced Reward

Dongheng Li[1], **Yongchang Hao**[1], **Lili Mou**[1,2]
[1]Dept. Computing Science & Alberta Machine Intelligence Institute (Amii), University of Alberta
[2]Canada CIFAR AI Chair, Amii
{dongheng,yongcha1}@ualberta.ca, doublepower.mou@gmail.com

Abstract

Large language models have become increasingly popular and demonstrated remarkable performance in various natural language processing (NLP) tasks. However, these models are typically computationally expensive and difficult to be deployed in resource-constrained environments. In this paper, we propose LLMR, a novel knowledge distillation (KD) method based on a reward function induced from large language models. We conducted experiments on multiple datasets in the dialogue generation and summarization tasks. Empirical results demonstrate that our LLMR approach consistently outperforms traditional KD methods on different tasks and datasets.

1. Introduction

Large language models (LLMs) have achieved remarkable performance in various text generation tasks, such as summarization (Ahmed and Devanbu, 2022; Nair et al., 2023) and dialogue systems (Deng et al., 2023; Cao et al., 2020). Moreover, this can be accomplished in a zero-shot manner, that is, a user enters a natural language prompt (e.g., "*Summarize the following text*") and the LLM will generate a desired output for the task (Brown et al., 2020). However, LLMs also present significant challenges. For example, the GPT-3 model has 175 billion parameters, which is resource-intensive, requiring significant computing power and memory. This might hinder real-world applications in resource-constrained environments.

Therefore, knowledge distillation (KD; Hinton et al., 2015) becomes an increasingly important research direction for LLMs (Gu et al., 2024; Wu et al., 2023; Hsieh et al., 2023), where the goal is to transfer the knowledge of LLM (called a *teacher*) to a smaller and more efficient model (called a *student*). Conventionally, this is accomplished by training the student from the teacher's predicted sentences or distributions (Kim and Rush, 2016). However, it has inherent limitations: during training, the student learns to predict the next word based on the teacher's previous predictions, whereas during inference, the student has to do so based on its own previous predictions. Such a discrepancy is known as *exposure bias*, and often leads to a performance degradation (Chiang and Chen, 2021; Ranzato et al., 2016).

In this paper, we propose a novel knowledge distilling method, based on reinforcement learning with a Large Language Model-induced Reward (dubbed LLMR). Instead of directly training from LLM's output, we first induce a q-value function from the LLM's policy (predicted probabilities) based on a widely adopted assumption (Chan and van der Schaar, 2021; Ramachandran and Amir, 2007; Ziebart et al., 2008), and then further induce a reward function based on the Bellman optimality equation (Sutton et al., 1999); this process follows our recent theoretical analysis between policies and rewards (Hao et al., 2022). The induced reward function is subsequently used to distill LLM's knowledge into the student, achieved by sampling a candidate sequence from the student-predicted distributions and evaluating it with the LLM-induced reward for policy gradient learning (Williams, 1992). In this way, our proposed LLMR distilling approach allows the student model to explore on its own during KD in a reinforcement learning (RL) fashion, thus alleviating the exposure bias problem.

We conducted experiments on two text generation tasks: dialogue generation and text summarization. Empirical results show that our LLMR approach largely outperforms traditional KD based on cross-entropy loss. We further quantitatively analyzed the exposure bias of the student models, verifying that RL indeed alleviates exposure bias arising during the KD process.[1]

2. Related Work

Knowledge distillation (KD) is effective in reducing the computing and memory demands of large neural networks while retaining high performance. Common KD approaches include matching output distributions (Hinton et al., 2015; Wu et al., 2023) and matching intermediate-layer representations (Romero et al., 2015; Polino et al., 2018; Sun et al., 2019).

[1]Our code is released as a GitHub repo: https://github.com/MANGA-UOFA/Prompt-LLMR

LREC-COLING 2024, pages 10657–10664
20-25 May, 2024.

KD has been applied to the sequence level for distilling text generation models (Kim and Rush, 2016; Wen et al., 2024) and autoregressive language models (West et al., 2022). Typically, the student learns from the teacher step by step with a cross-entropy loss, but such an approach may suffer from exposure bias (Ranzato et al., 2016). Researchers have proposed reverse Kullback–Leibler (Tu et al., 2020; Gu et al., 2024) and generalized f-divergence (Wen et al., 2023b) losses, which involve student sampling but still follow the spirit of traditional KD pushing the student's distribution to the teacher's step by step. In our LLMR method, on the other hand, the teacher only scores a student-sampled sequence, which allows more exploration during the KD process.

Reinforcement learning (RL) has been widely used for text generation, especially for alleviating exposure bias (Ranzato et al., 2016; Gu et al., 2024). A key design choice is the reward function, which in previous work is often given by task heuristics with groundtruth sequences (Sokolov et al., 2016; Pang and He, 2021) or trained reward models (Bahdanau et al., 2017; Paulus et al., 2018). Our LLMR method follows previous theoretical work (Hao et al., 2022), but directly induces a reward function from a pretrained LLM in a principled and task-agnostic manner.

3. Approach

Problem Formulation. Knowledge distillation (KD) aims to transfer the knowledge of a teacher model to a student. Although the student can solely learn a task from a parallel corpus $D_p = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{M}$, it is argued that the teacher's predicted distribution contains more knowledge than an annotated label $\mathbf{y}$ (Hinton et al., 2015). Kim and Rush (2016) propose SeqKD and minimize a Kullbuck–Leibler loss, equivalent to minimizing a cross-entropy loss, at the sequence level between a teacher p and a student q_θ by $J_{\text{SeqKD}} = \mathbb{E}_{\mathbf{y}\sim p}\left[\log \frac{p(\mathbf{y}|\mathbf{x})}{q_\theta(\mathbf{y}|\mathbf{x})}\right]$. In practice, the expectation over the sentence space is intractable. To tackle this, they use a hard sequence $\mathbf{y}$ generated by beam search on the teacher model as an approximation: $\hat{J}_{\text{SeqKD}} = -\log q_\theta(\mathbf{y}|\mathbf{x})$.

In our work shown in Figure 1, we prompt a large language model (LLM) and treat it as the teacher. However, we do not follow the common KD that minimizes the divergence between LLM's probability p_{LLM} and the student q_θ. Instead, we propose to induce a reward function R_{LLM} from p_{LLM} and adopt reinforcement learning for KD with objective:

$$\text{maximize}_\theta \, \mathbb{E}_{\mathbf{y}\sim q_\theta}[R_{\text{LLM}}(\mathbf{y})] \quad (1)$$

Our approach alleviates the exposure bias problem (Chiang and Chen, 2021; Ranzato et al., 2016) in traditional KD, where the student is fed with the teacher's predicted prefix during training, but only has access to its own partial prediction during inference. By contrast, our RL-based KD allows the student to explore with its own predicted sequence, shown by $\mathbf{y} \sim q_\theta$ in (1), which bridges the gap between training and inference.

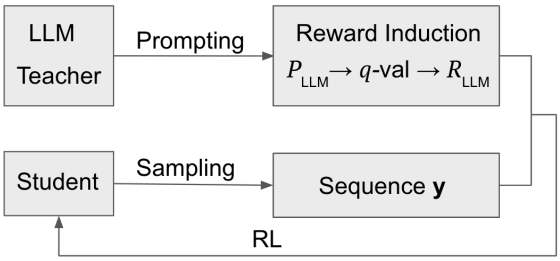

Figure 1: Overview of the approach.

In the rest of this section, we will introduce the reward R_{LLM} and the optimization of (1) in detail.

Inducing Reward from LLMs. We propose to induce a reward function from large language models (LLMs) for RL-based KD, inspired by the theoretical analysis that links policies (predicted probabilities) and reward functions (Hao et al., 2022). In our work, we design an intuitive prompt to obtain the LLM's policy for reward induction.

Consider a task $\mathcal{T}$ and an input sentence $\mathbf{x}$. We formulate a prompt as $\texttt{pmt}_{\mathcal{T}}(\mathbf{x})$. In fact, the prompt depends on the task of interest, and in our experiments, two common text generation tasks are considered: summarization (Ahmed and Devanbu, 2022; Nair et al., 2023) and dialogue generation (Deng et al., 2023; Cao et al., 2020). Our prompts are

$$\texttt{pmt}_{\text{sum}}(\mathbf{x}) \equiv \text{“Summarize [} \mathbf{x} \text{]:”}$$
$$\texttt{pmt}_{\text{dialog}}(\mathbf{x}) \equiv \text{“The dialogue response of [} \mathbf{x} \text{] is:”}$$

where $\mathbf{x}$ is the original input sentence and the square brackets are delimiters specifying the input boundaries.

Given a candidate output $\mathbf{y} = (\text{y}_1, \cdots, \text{y}_T)$, our goal is to induce a reward function $R_{\text{LLM}}(\mathbf{y})$ that evaluates the "goodness" of $\mathbf{y}$. This requires modeling text generation as a Markov decision process (MDP), where an action is the prediction of the next word and a state is the partially predicted sequence in addition to the prompt. The state transition is a deterministic process that simply appends the newly generated word to the previous state.

Our reward induction starts by querying an LLM in a step-by-step fashion to obtain the next word probability $p_{\text{LLM}}(\text{y}_t|\mathbf{y}_{<t}, \texttt{pmt}_{\mathcal{T}}(\mathbf{x}))$. Notice that we do not let the LLM generate outputs during RL-based KD, but the prefix $\mathbf{y}_{<t}$ and the next word y_t are from the student-sampled sequence. The role

of LLM is to predict its probability and to induce a reward for $\mathbf{y}$.

With the next-word probability, we are able to induce a q-value function for step t, which indicates the goodness of an action, i.e., the word y_t, at the state $(\mathbf{y}_{<t}, \mathrm{pmt}_{\mathcal{T}}(\mathbf{x}))$. The q-value induction process is based on the common assumption (Chan and van der Schaar, 2021; Ramachandran and Amir, 2007; Ziebart et al., 2008) that an action is taken stochastically based on a Boltzmann distribution induced by q-values:

$$p_{\text{LLM}}(\mathrm{y}_t|\mathbf{y}_{<t}, \mathrm{pmt}_{\mathcal{T}}(\mathbf{x})) = \frac{\exp\{q\text{-val}(\mathrm{y}_t; \mathbf{y}_{<t})\}}{\sum_{\mathrm{y}'} \exp\{q\text{-val}(\mathrm{y}'; \mathbf{y}_{<t})\}} \quad (2)$$

where the q-value function also depends on $\mathrm{pmt}_{\mathcal{T}}(\mathbf{x})$ but is omitted for simplicity.

In other words, the assumption implies that a higher-valued action will be taken with a larger probability, which makes much sense in practice. Moreover, the resemblance between (2) and a softmax function suggests that we may directly take the LLM's logit (pre-softmax value) f_{LLM} as the q-value in the MDP modeling.

The final step of reward induction is based on Bellman optimality (Degris et al., 2012; Sutton and Barto, 2018), which derives an optimal q-value function from a reward. We follow the practice of inverse reinforcement learning (Ramachandran and Amir, 2007; Ziebart et al., 2008; Chan and van der Schaar, 2021) and use Bellman optimality in an opposite way to derive a reward R_{LLM} from the q-value function in (2):

$$\begin{aligned} R_{\text{LLM}}(\mathrm{y}_t; \mathbf{y}_{<t}) =& q\text{-val}(\mathrm{y}_t; \mathbf{y}_{<t}) \\ & - \max_{\mathrm{y}'} q\text{-val}(\mathrm{y}'; \mathbf{y}_{<t+1}) \end{aligned} \quad (3)$$

In this way, our derived reward R_{LLM} evaluates the appropriateness of a word y_t at every step given its context $\mathbf{y}_{<t}$. That is to say, such a reward function is dense as opposed to various other heuristic rewards (e.g., BLEU scores) that only come at the end of a sequence (Wu et al., 2017). The overall reward induction process follows our previous work (Hao et al., 2022), but this paper extends it to a new scenario. Hao et al. (2022) train a sequence-to-sequence network in a supervised manner on a parallel corpus and perform semi-supervised learning on non-parallel corpora. Our paper shows that a reward function can be derived directly from a pretrained LLM and helps various text generation tasks, which is a new insight, especially in the LLM era.

Reinforcement Learning-Based KD. Our derived reward function allows us to perform reinforcement learning (RL) for KD. Specifically, a sequence is sampled from the student's prediction, given by $\mathbf{y} \sim q_\theta$. Then, each word y_t in $\mathbf{y}$ is evaluated by the induced reward function (3), and our total reward of the sequence is

$$R_{\text{LLM}}(\mathbf{y}) = \sum\nolimits_t R_{\text{LLM}}(\mathrm{y}_t; \mathbf{y}_{<t}) \quad (4)$$

which is our objective to maximize, as shown in Eqn. (1).

Since the parameter θ occurs during the sampling process, the gradient cannot be obtained by backpropagation, and RL is required to train θ in a trial-and-error manner. In NLP, the REINFORCE method is commonly used (Ranzato et al., 2016; Wang et al., 2020), where the gradient is given by

$$\nabla_\theta \mathop{\mathbb{E}}_{\pi_\theta}\left[\sum_t R_{\text{LLM}}(\mathrm{y}_t; \mathbf{y}_{<t})\right] = \mathop{\mathbb{E}}_{\pi_\theta}\left[\sum_t G_t(\mathbf{y}) \log \pi_\theta(\mathrm{y}_t; \mathbf{y}_{<t})\right]$$

where $G_t(\mathbf{y})$ is known as the gain in the RL literature, being the accumulated reward from step t, given by $G_t(\mathbf{y}) := \sum_{\tau \geq t} R_{\text{LLM}}(\mathrm{y}_\tau; \mathbf{y}_{<\tau})$.

Overall, our RL-based KD differs from traditional sequence-level KD, where the teacher teaches unilaterally with its own prediction, i.e., $\mathbf{y} \sim p_{\text{LLM}}$. Instead, we allow the student to generate its own prediction, and the LLM teaches by evaluating the "goodness" of the student's output. In this way, our approach alleviates the exposure bias problem, as the student is aware of its own partial prediction during training. Compared with classic RL-based text generation, we do not require heuristically designed reward functions (Bahdanau et al., 2017; Shen et al., 2016) or human feedback reward models (Ouyang et al., 2022; Ziegler et al., 2019).

4. Experiments

Setups. We evaluated our approach on two text generation tasks with three datasets: DailyDialog (Li et al., 2017) and OpenSubtitles (Lison and Tiedemann, 2016) for dialogue generation, as well as CNN/DailyMail (See et al., 2017; Hermann et al., 2015) for summarization. In particular, dialogue datasets tend to have sample-overlapping issues between training and test sets, and we used the cleaned version (Wen et al., 2022) for rigorous experimentation.

Our teacher was a T0-3B model (Sanh et al., 2022) and the student was T5-Base with 220 million parameters (Raffel et al., 2020). Since our RL-based KD requires meaningful sampling from the student, we performed pre-distillation by the standard cross-entropy loss (Kim and Rush, 2016), which is common in KD research (Wen et al., 2023b; Shleifer and Rush, 2020) and shows our method provides add-on improvement.

It should be emphasized that our work addresses unsupervised KD, where the training process only used unlabeled input sentences without groundtruth references. During validation and test

	Model		DailyDialog		OpenSubtitles		CNN/DailyMail		
			BLEU2	BLEU4	BLEU2	BLEU4	ROUGE-1	ROUGE-2	ROUGE-L
1	Prompting Teacher		5.57	1.49	4.67	1.51	36.16	14.99	24.05
2	Prompting Student		1.35	0.31	1.21	0.25	21.23	6.73	17.88
3	Distilled Students	SeqKD	6.19	1.71	3.87	1.35	35.46	14.52	23.68
4		KL	5.03	1.40	3.84	1.33	34.11	14.21	22.83
5		RKL	5.02	1.29	4.12	1.36	32.07	13.77	22.87
6		JS	6.60	1.73	3.64	0.87	35.88	14.72	23.97
7		Our LLMR	**7.00**	**1.88**	**5.13**	**1.85**	**36.42**	**15.21**	**24.83**

Table 1: Main results on dialogue generation and summarization tasks.

phases, the ground truths were used in the standard evaluation metrics: BLEU (Papineni et al., 2002) for dialogue generation and ROUGE (Lin, 2004) for summarization.

Main Results. Table 1 presents the performance of our model and baselines. As seen, the teacher model (Row 1) achieves decent performance in these tasks. The results are slightly lower than, or comparable to, those of supervised methods reported in previous literature, for example, 8.96 BLEU2 for DailyDialog (Hao et al., 2022) and 39.5 ROUGE-1 for CNN/DailyMail (Vaswani et al., 2017). This is understandable because our teacher is directly prompted for the tasks without finetuning. On the other hand, prompting the student (Row 2) does not yield meaningful performance, which is consistent with the findings of the scaling effect (Kim and Rush, 2016; Hinton et al., 2015; Wen et al., 2023b). The strong teacher and weak student jointly set up a reasonable foundation for our distillation research.

Rows 3–7 present the performance of different distilling methods, showing that KD can indeed transfer the teacher's knowledge into the student. Among different KD methods, SeqKD (Kim and Rush, 2016) employs hard samples to train the student, and achieves close performance to the teacher; in particular, it surpasses the teacher on DailyDialog, which can be interpreted by smoothing the noise of the teacher (an unfinetuned prompting system). We also experimented with soft distillation based on various f-divergence functions, including Kullback–Leibler (KL), Reverse KL (RKL), and Jenson–Shannon (JS) divergences (Wen et al., 2023b). As seen, the results are not fully consistent, although JS tends to perform better in general.

Our LLMR (Row 7) performs reinforcement learning based on a reward function induced from the teacher model. It achieves superior performance across all the metrics and datasets, consistently demonstrating the effectiveness of our approach.

Diversity Analysis. The diversity of output text is considered an important aspect of text generation systems (Li et al., 2016; Wen et al., 2023a). We evaluated the diversity of competing models by the standard distinct n-gram measures (Li et al., 2016; Pang and He, 2021; Ji et al., 2023), given by

$$\text{Distinct-}n = \frac{\text{Number of unique } n\text{-grams}}{\text{Total number of } n\text{-grams}}$$

Model	DailyDialog		OpenSubtitles		CNN/DailyMail	
	Dist1	Dist2	Dist1	Dist2	Dist1	Dist2
SeqKD	4.93	27.37	4.78	23.15	3.86	33.59
KL	4.76	26.77	4.99	24.00	3.76	33.59
RKL	5.76	29.01	5.38	23.72	4.07	32.27
JS	5.84	32.25	4.44	19.21	3.83	31.47
Our LLMR	**6.02**	**34.83**	**5.82**	**27.21**	**4.20**	**35.38**

Table 2: Distinct n-gram (Dist$n$) scores.

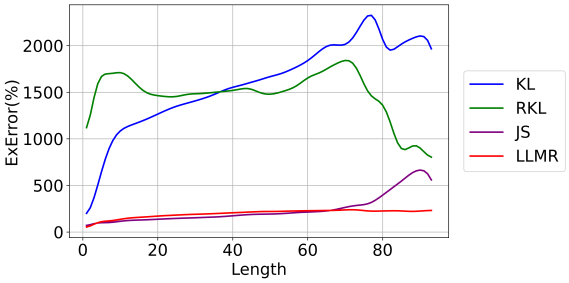

Figure 2: The averaged excess error (ExError) with respect to sequence length of different models on DailyDialog.

As seen in Table 2, the KL loss achieves low distinct scores, which is consistent with previous evidence that the KL training makes the model generate dull and short utterances (Wei et al., 2019; Wen et al., 2023a). By contrast, our LLMR yields much higher distinct scores, which verifies that our RL mechanism allows the model to explore different regions of the sentence space, leading to much more diverse output.

Exposure Bias Analysis. As mentioned in §1, our LLMR adopts RL and is supposed to alleviate exposure bias during KD. We quantify the amount of exposure bias by adapting a recently established measure, Excess Error Percentage (ExError%, Arora et al., 2022). In our scenario, ExError% is defined by

$$\text{ExError\%}(l) = \frac{D_s(l) - D_t(l)}{D_t(l)} \times 100\%$$

Here, $D_s(l)$ stands for the accumulated Kullback–Leibler (KL) divergence between the teacher and student, when the models follow the student's trajectory up to the $(t-1)$th step:

$$D_s(l) = \sum_{t=1}^{T} \mathbb{E}_{\substack{\mathsf{y}_{<t} \sim q_\theta(\cdot|\mathsf{x}) \\ \mathsf{y}_t \sim p(\cdot|\mathbf{y}_{<t}, \mathsf{x})}} \left[\log \frac{p(\mathsf{y}_t|\mathbf{y}_{<t}, \mathsf{x})}{q_\theta(\mathsf{y}_t|\mathbf{y}_{<t}, \mathsf{x})} \right]$$

whereas $D_t(l)$ is the KL divergence when the models follow the teacher's trajectory up to the $(t-1)$th step:

$$D_t(l) = \sum_{t=1}^{T} \mathbb{E}_{\substack{\mathsf{y}_{<t} \sim p(\cdot|\mathsf{x}) \\ \mathsf{y}_t \sim p(\cdot|\mathbf{y}_{<t}, \mathsf{x})}} \left[\log \frac{p(\mathsf{y}_t|\mathbf{y}_{<t}, \mathsf{x})}{q_\theta(\mathsf{y}_t|\mathbf{y}_{<t}, \mathsf{x})} \right]$$

Overall, ExError% measures the percentage of excess error when the models follow the student's trajectory, compared with following the teacher's trajectory. Typically, ExError% is positive and a higher value indicates more exposure bias. It can go over 100% because the KL divergence is not upper bounded.

As seen in Figure 2, KL- and RKL-based KD methods yield high exposure bias, which is expected as the KL and RKL divergence functions are asymmetric and do not push the student to the teacher well. The JS divergence is symmetric and JS-based KD requires both teacher and student samplings. Its ExError% remains low at the beginning, but grows when the sequence becomes longer. Our LLMR approach employs RL training and achieves low ExError% throughout different lengths. The experiment confirms our approach alleviates exposure bias and explains the performance improvement in main results.

5. Conclusion

In this paper, we propose a novel knowledge distillation method, called LLMR, based on a large language model-induced reward function. Experiments on dialogue generation and text summarization show that our approach outperforms previous KD methods in terms of various metrics. We also conducted a detailed analysis to verify that our reinforcement learning-based method indeed alleviates the exposure bias problem present in common KD approaches.

6. Acknowledgments

We thank all reviewers and chairs for their valuable comments. The research is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No. RGPIN2020-04465, an Alberta Innovates Project, the Amii Fellow Program, the Canada CIFAR AI Chair Program, a UAHJIC project, a donation from DeepMind, and the Digital Research Alliance of Canada (alliancecan.ca).

7. Bibliographical References

Toufique Ahmed and Premkumar Devanbu. 2022. Few-shot training LLMs for project-specific code-summarization. In *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*.

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Cheung. 2022. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL*, pages 700–710.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *International Conference on Learning Representations*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901.

Yu Cao, Wei Bi, Meng Fang, and Dacheng Tao. 2020. Pretrained language models for dialogue generation with multiple input sources. In *Findings of the Association for Computational linguistics: EMNLP*, pages 909–917.

Alex J Chan and Mihaela van der Schaar. 2021. Scalable Bayesian inverse reinforcement learning. In *International Conference on Learning Representations*.

Ting-Rui Chiang and Yun-Nung Chen. 2021. Relating neural text degeneration to exposure bias. In *BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 228–239.

Thomas Degris, Martha White, and Richard S Sutton. 2012. Off-policy actor-critic. In *International Conference on Machine Learning*.

Yang Deng, Wenqiang Lei, Lizi Liao, and Tat-Seng Chua. 2023. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. In *Findings of the Conference on Empirical Methods in Natural Language Processing*, page 10602–10621.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*.

Yongchang Hao, Yuxin Liu, and Lili Mou. 2022. Teacher forcing recovers reward functions for text generation. In *Advances in Neural Information Processing Systems*, pages 12594–12607.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Conference of the Association for Computational Linguistics: ACL*, pages 8003–8017.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55:1–38.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the International Joint Conference on Natural Language Processing*, pages 986–995.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 923–929.

Varun Nair, Elliot Schumacher, Geoffrey Tso, and Anitha Kannan. 2023. DERA: Enhancing large language model completions with dialog-enabled resolving agents. *arXiv preprint arXiv:2303.17071*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, pages 27730–27744.

Richard Yuanzhe Pang and He He. 2021. Text generation by learning from demonstrations. In *International Conference on Learning Representations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Antonio Polino, Razvan Pascanu, and Dan Alistarh. 2018. Model compression via distillation and quantization. In *International Conference on Learning Representations*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Deepak Ramachandran and Eyal Amir. 2007. Bayesian inverse reinforcement learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2586–2591.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for thin deep nets. In *International Conference on Learning Representations*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1683–1692.

Sam Shleifer and Alexander M Rush. 2020. Pre-trained summarization distillation. *arXiv preprint arXiv:2010.13002*.

Artem Sokolov, Julia Kreutzer, Stefan Riezler, and Christopher Lo. 2016. Stochastic structured prediction under bandit feedback. In *Advances in Neural Information Processing Systems*, pages 1489–1497.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 4323–4332.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*.

Lifu Tu, Richard Yuanzhe Pang, Sam Wiseman, and Kevin Gimpel. 2020. ENGINE: Energy-based inference networks for non-autoregressive machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 2819–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. 2020. Neural policy gradient methods: Global optimality and rates of convergence. In *International Conference on Learning Representations*.

Bolin Wei, Shuai Lu, Lili Mou, Hao Zhou, Pascal Poupart, Ge Li, and Zhi Jin. 2019. Why do neural dialog systems generate short and meaningless replies? a comparison between dialog and translation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7290–7294.

Yuqiao Wen, Yongchang Hao, Yanshuai Cao, and Lili Mou. 2023a. An equal-size hard em algorithm for diverse dialogue generation. In *International Conference on Learning Representations*.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023b. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 10817–10834.

Yuqiao Wen, Guoqing Luo, and Lili Mou. 2022. An empirical study on the overlapping problem of open-domain dialogue datasets. In *Proceedings of the Language Resources and Evaluation Conference*, pages 146–153.

Yuqiao Wen, Behzad Shayegh, Chenyang Huang, Yanshuai Cao, and Lili Mou. 2024. EBBS: An ensemble with bi-level beam search for zero-shot machine translation. *arXiv preprint arXiv:2403.00144*.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: From general language models to commonsense models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Language*, 8:229–256.

Lijun Wu, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2017. Sequence prediction with unlabeled data by reward function learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3098–3104.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2023. LaMini-LM: A diverse herd of distilled models from large-scale instructions. *arXiv preprint arXiv:2304.14402*.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1433–1438.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.