ARLHF: a Framework to Enhance RLHF by Active Learning on NLP Tasks

Anjie Liu* Qi WANG* Guangzheng Xu* Jieming Zhang*

April 3, 2023

Abstract

Active learning (AL) is a widely-used technique in supervised learning tasks to mitigate the costs and time associated with labeling data. Reinforcement learning for human feedback (RLHF) is a novel approach to align language models with human preferences. In this study, we propose a new framework, ARLHF[1], which augments conventional RLHF with active learning. The use of active learning is expected to alleviate the challenge of collecting high-quality human feedback, which is often a laborious and time-consuming process. Our results, however, did not demonstrate a clear advantage of active learning in the task of text summarization, largely due to the considerable cost of tuning each model in the workflow. Future work could concentrate on addressing the limitations of active learning in large datasets and refining the selection of hyperparameters to enhance the efficiency and accuracy of text summarization.

1 Introduction

As the volume of data grows exponentially, text summarization has become a crucial task in natural language processing. In addition, to effectively leverage certain non-differentiable metric scores in NLP, we have employed reinforcement learning to fine-tune our large language model. However, collecting high-quality human feedback is the most crucial aspect, but it requires a substantial amount of training data, which can be expensive and time-consuming to annotate. To address this issue, we propose a new framework, ARLHF, to combine active learning with reinforcement learning fine-tuning of a language model to improve text summarization.

Our approach employs active learning to sample the most valuable data points from a large dataset containing articles and corresponding human feedback for different summaries. We use these data points to train a reward model for fine-tuning our language model by reinforcement learning method. It allows us to increase the the probability of producing text that is similar to that written by humans and generate outputs of superior quality, as evaluated by humans.

Compared to traditional text summarization methods, our approach has several advantages. Firstly, it reduces annotation costs by selecting the most informative data points for annotation, which can improve the efficiency of the summarization process. Secondly, RL fine-tuning enables us to directly optimize summarization performance, leading to higher quality summaries. Finally, our approach is generalizable and can be applied to different summarization tasks and domains.

2 Related Work

Active Learning. Due to the limitations of computational resources, we implement active learning to achieve optimal performance when fine-tuning large language models through reinforcement learning with human feedback in limited data. Currently, two distinct methods are used for active learn-

*These authors contributed equally to this work.

[1] https://github.com/vcvcvnvcvcvn/myrl4lm

ing: query-acquiring (pool-based) and query-synthesizing. Pool-based techniques are used to identify the most valuable and informative samples through a variety of query strategies. On the other hand, query-synthesizing approaches leverage generative models to create data and distinguish between the most informative samples by utilizing an adversarial network to differentiate between original and synthesized data (Sinha et al., 2019). In our experiment, we employ the query-acquiring method to classify text preferences and identify the most significant samples for the entire dataset, apply these selected data in our following summarization model.

When utilizing the pool-based algorithm in active learning, selecting the appropriate query strategy to identify the samples to be labeled is critical for achieving success. Numerous sampling strategies have been proposed to process binary classification transformer models. Among these, confidence-based strategies such as Least Confidence, Prediction Entropy, Breaking Ties (Luo et al., 2005), Bayesian Active Learning by Disagreement (BALD) (Houlsby et al., 2011), and Contrastive Active Learning (Margatina et al., 2021) are widely used. Additionally, coreset strategies, including Greedy Coreset (Sener and Savarese, 2018) and Lightweight Coreset (Bachem et al., 2018), have also demonstrated strong performance. Furthermore, embedding-based strategies such as embedding k-means, Discriminative Active Learning (Gissin and Shalev-Shwartz, 2019), and Efficient Active Learning and Search (SEALS) (Coleman et al., 2021) have been developed and are gaining attention in our experiment.

Reinforcement learning for language model. Pretrained language models can be fine-tuned with small datasets to adapt to downstream tasks. In addition to supervised fine-tuning, reinforcement learning has emerged as an alternative method to fine-tune language models. This approach enables the optimization of non-differential metrics such as BLEU and ROUGE to improve the parameters in the model. Fine-tuning models directly with supervised learning, assuming that reference summaries are given, can lead to the issue of "exposure bias" (Schmidt, 2019). As such, reinforcement learning has been utilized in many natural language processing tasks, including text summarization (Paulus et al., 2017), machine translation (Kiegeland and Kreutzer, 2021), dialogue (Saleh et al., 2020) to avoid the risk of exposure bias.

Recently, Ouyang et al. (2022) managed to fine-tune GPT-3 with Proximal Policy Optimization (PPO) to align the model with human preference. The resulting 1.3B parameter GPT-3, after fine-tuning with human feedback, outperforms the 175B GPT-3. Although reinforcement learning-based fine-tuning has shown significant improvements, Choshen et al. (2019) have pointed out that it is significantly less stable than supervised learning.

3 Method

3.1 Framework

After selecting a downstream task, collecting data for the task, the entire framework for ARLHF can be divieded into 3 steps.

Step 1: Use an active learning model to select data points for labeling by human annotators. The goal is to select the most informative data points for labeling to minimize the amount of human effort required. Then a reward model will be trained to predict the human-preferred output.

Step 2: The reward model is trained further using the labeled data picked by the active learning model to improve its accuracy in predicting the quality of generated text.

Step 3: The reward model is used to fine-tune the language model for the downstream task using the PPO algorithm. The output of the reward model serves as a scalar reward signal for the PPO algorithm to optimize against. The goal is to train a language model that generates text that maximizes the reward signal from the reward model.

Steps 1 and 2 can be iterated continuously until the reward model reaches a promising accuracy.

3.2 Active Learning

Active learning is an algorithm to select the most informative data points to label, rather than labeling all data points in the training set. This helps to reduce the amount of labeled data required for training a model, then save the computational resources and often lead to better model performance with less data (Shown as table below). In this report, we used the Query-by-Committee method which selects instances that are the most uncertain or where there is disagreement among a committee of models. The idea is that if multiple models disagree on the label of a data point, then it is likely to be an informative point to label.

The query strategy that we employed, called “Breaking Ties”, which has the best performance in our experiment, is based on the utilization of soft margin support vector machines (SVMs), to select samples based on the criterion of having a minimal difference between the probability of the predicted class with the highest likelihood and that of the second most probable class (Luo et al., 2005). This method uses a one-vs-one approach, to enhance the confidence of the multi-class classification through probability estimation. In this method, the SVMs transform the data points into a higher dimensional space using the function $\phi(x)$, which results in the use of a hyperplane to separate the data into two distinct classes. To avoid the need for explicit calculations of the inner product in the high-dimensional feature space, a kernel $k(x,y) = \langle \phi(x), \phi(y) \rangle$ is typically employed.

Given the input $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n} \in R^n \times \{-1, 1\}$ be a training set. Ideally, we would like to

$$\text{minimize } \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\zeta_i$$

subject to

$$y_i(\mathbf{w}^T\phi(x_i) + b) \leq 1 - \zeta_i \text{ and } \zeta_i \geq 0$$

for $i = 1, ..., n$ where C is a scalar value used to control the balance between the margins $\frac{2}{||\mathbf{w}||}$ and the empirical risk, the slack variables ζ_i relax the separation constraints.

Then we can solve the constraint optimization problem with a Lagrange multiplier α_i. The decision function, which also the separating hyperplane, is

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b$$

This is how the Breaking Ties method is utilized for binary classification. Although this technique is capable of accommodating multi-class classification, our experiment only required binary SVMs. Therefore, we did not delve into the details of using multi-class SVMs in this context.

3.3 Reinforcement learning for text generation

3.3.1 Problem Definition

The task of text generation can be formulated as a reinforcement learning problem. In this context, the generated text can be viewed as a statement, and the neural network can be used to train a parameterized policy function π_θ to maximize the long-term discounted rewards of a trajectory $E_\pi[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)]$. As a result, policy gradient methods can be utilized to optimize the parameters during the model fine-tuning process. The most commonly used gradient estimator is

$$\hat{E}_\pi \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta \left(a_t \mid s_t \right) \hat{A}_t \right]$$

Here, $\hat{A}_t$ is an alternative reward signal that removes the baseline reward to stabilize the training process.

3.3.2 Proximal Policy Optimization

Schulman et al. (2017) introduced Proximal Policy Optimization (PPO) as a method to use off-policy gradient descent for optimizing deep neural networks. In gradient policy methods, it is inefficient to sample a large number of trajectories because previous samples become irrelevant when the policy parameters are updated. While importance sampling weights can be used, they result in

```
Active Learning Algorithm
Input Initial labeled training dataset $\mathcal{L}$ and a labeled data pool $\mathcal{U}$
For $i = 1, 2, ...$ do
    Train a classifier $f$ from the training dataset $\mathcal{L}$
    Select samples $x^* \in \mathcal{U}$ by the certain query strategy according to $f$
    Obtain the corresponding $y^*$ from $\mathcal{U}$
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x^*, y^*)\}$
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \{(x^*, y^*)\}$
end for
```

high variance. To address this trade-off between off-policy and low efficiency, PPO proposes a surrogate objective function to optimize, which has the following form:

$$E_\pi[\sum_{t=0}^{T} \min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t))]$$

Here, $r_t(\theta) = \frac{\pi_\theta(s_t,a_t)}{\pi_{\theta'}(s_t,a_t)}$ represents the importance sampling weights. The surrogate objective function is designed to prevent the updated policy network from being too different from the previous one.

3.3.3 Reward Model

Here, we have a Reddit post and two summaries (chosen and rejected) as input. The ground truth label (labels) is the human feedback (0 for chosen and 1 for rejected). And the loss function is given as:

$$loss(r_\theta) = -E[log(\sigma(r_\theta(x, y_i)) - r_\theta(x, y_{1-i}))]$$

In the above formulation, y_i where i $\in \{0, 1\}$ is a human preferred or chosen summary. The reward model r_θ takes the post x and the summary y and return a scalar value. The value is computed for both the summaries and a σ activation is applied to the difference. Finally, the negative log is computed.

4 Experiment

4.1 Active Learning

To perform preference classification, we utilized an outperforming pre-trained model MPNet (Song et al., 2020), which is a noval combination of the Masked Language Modeling (MLM) and Permuted Language Modelling (PLM) techniques. The structure is shown as Figure 1. This model combines the strengths of BERT (Devlin et al., 2019) and XLNet (Yang et al., 2020) while simultaneously addressing their limitations. MPNet considers the relationships among predicted tokens, and it efficiently integrates the complete positional information of a sentence, thus avoiding any position inconsistencies between pre-training and fine-tuning.

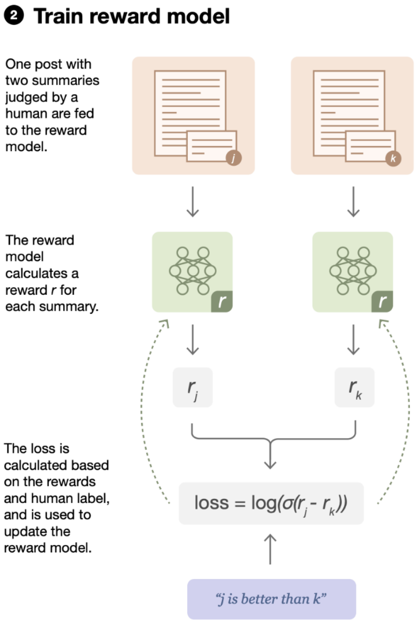
2 Train reward model
One post with two summaries judged by a human are fed to the reward model.
j
k
The reward model calculates a reward r for each summary.
r
r
r_j
r_k
The loss is calculated based on the rewards and human label, and is used to update the reward model.
loss = log(σ(r_j - r_k))
"j is better than k"

Figure 1: Train reward model

We have employed the sentence-transformer (Reimers and Gurevych, 2019) model in our experiment, which is a variation of transformer-based models designed to generate semantically significant sentence embeddings. We have compared these embeddings using various similarity

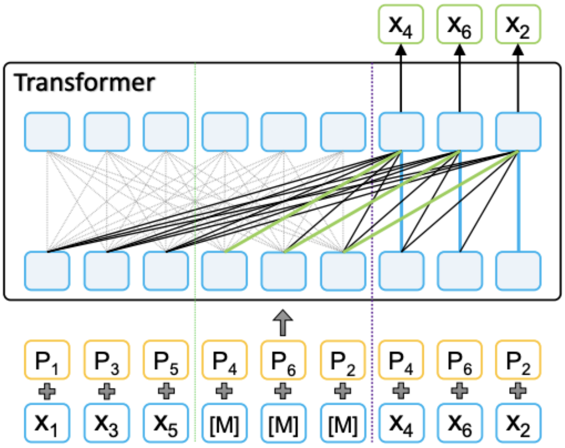

Figure 2: The architecture of MPNet. x_i and p_i denote the token and position embeddings respectively. The non-predicted part of the architecture is represented by light gray lines that indicate bidirectional self-attention. The attention mask in query streams and content in two-stream self-attention are respectively represented by green and blue lines, and black lines represent the overlapped of blue and green lines. (Song et al., 2020)

measures. While the primary purpose of sentence-transformer models is semantic similarity search and clustering, they are not ideal for text classification tasks. Nonetheless, due to their superior computational speed over transformer-based models, we have utilized the sentence-transformer fine-tuning (SETFIT) (Tunstall et al., 2022) to fit the pre-trained model in the down-streaming classification tasks.

SETFIT adopts a two-part training method shown as the Figure 2. Firstly, it adjusts a sentence-transformer to the input data by utilizing a contrastive, Siamese approach on pairs of sentences. Secondly, it trains a classifier head using the encoded training data produced from the first step (Tunstall et al., 2022). As a result, we can classify text preferences with similar optimal performance at a highly accelerated speed.

In this part, we apply the “small-text” (Schröder et al., 2023) library to analyze the performance of active learning. At the beginning, we have used random sampling as a baseline to determine the best performing query strategy among all possible methods. Due to constraints in computational resources, we have only conducted tests on a smaller dataset consisting of one thousand training examples and one hundred test instances. We have initialized our experiment with 40 labeled data points, and in each iteration, the program requests an additional 20 labels from the pool of unlabeled data. As a result, we were able to obtain a total of 140 meaningful labeled data points. When comparing different query strategies, it is evident that the Breaking Ties strategy outperforms other methods. Similarly, the Greedy Coreset achieves a higher test accuracy compared to random sampling. While Lightweight Coreset and Contractive Active Learning may outperform random selection, the limited number of iterations and the uncertainty of sampling prevent us from drawing a conclusive result with high confidence and strong evidence. In contrast, other approaches did not perform well on our dataset.

4.2 Reward Model

The input summaries (a chosen summary and a rejected summary) are passed through a pre-trained GPT language model to obtain hidden states for each token in the summaries. The hidden states are fed into a linear layer to obtain a scalar reward value for each token in the summaries. The scalar reward values are obtained by multiplying the hidden state of each token by a weight matrix and then applying a linear activation function. The reward values for the chosen summary and the rejected summary are sliced based on the point where they start to diverge. This is done by finding the index of the first occurrence where the chosen summary input ids and the rejected summary input ids are different. The reward values for the chosen summary and the rejected summary are further truncated based on the location of the first padding token. The padding token is used to indicate the end of a summary. By truncating the reward values at the location of the first padding token, we ignore any reward values that might have been generated for the padding tokens.

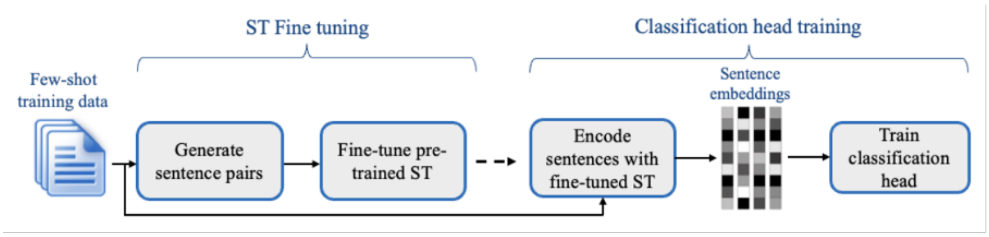

Figure 3: SETFIT's fine-tuning and training block diagram (Tunstall et al., 2022)

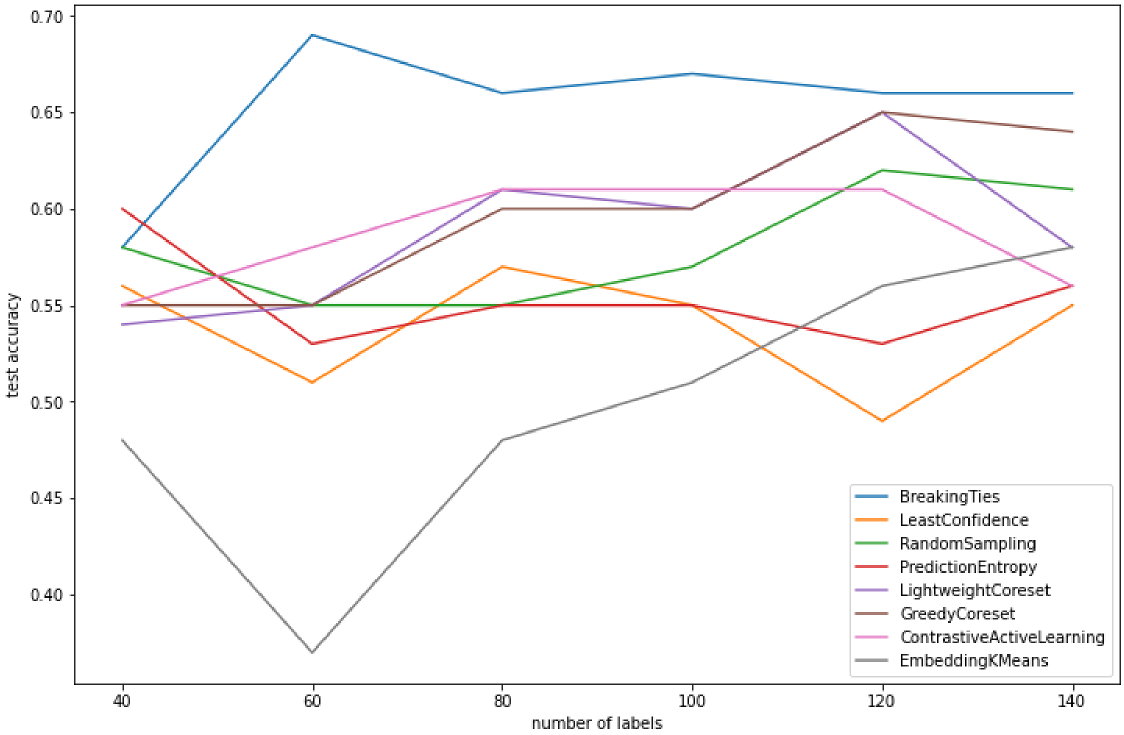

Figure 4: Comparison between different strategies

The final reward score is taken as the reward value for the last token in the chosen summary. This is because the last token is typically the most important token in the summary, as it provides the most information about the content of the summary.

Correct number plus one if the reward of chosen summary is greater than the reward of rejected summary in test data, and the test accuracy is correct number divide by total number of test data. The test accuracy of the reward model trained with 10000 random data is 0.5698, and the test accuracy of the reward model trained with 10000 data selected by active learning is 0.4354.

4.3 End-to-end experiments on text summarization task

Text summarization is a challenging task in natural language processing, and the CNN/Daily Mail (CNN/DM) dataset is widely used for abstractive summarization. Table 1 presents the data statistics of this dataset.

Data Split			Average #Tokens	
Train	Vali	Test	Source	Reference
287K	13K	11K	384.0	61.3

Table 1: CNN/DM dataset statistics

In this study, we used the **rl4lm** package developed by Ramamurthy et al. (2022) for end-to-end training. The package integrates

PPO algorithm and supervised learning to fine-tune a pre-trained language model. We trained a reward model ourselves to obtain the reward during reinforcement learning fine-tuning.

We chose the T5 model developed by Raffel et al. (2020) as the pre-trained model due to its impressive performance on various NLP tasks. Specifically, we used t5-small with 60 million parameters due to the limited computing resources. For the end-to-end experiments, the reward model was further trained with 10000 data picked randomly or by active learning. Additionally, we fine-tuned the model using supervised learning, ROUGE-PPO, and two different reward models mentioned above. The supervised-learning fine-tuning served as the baseline, and ROUGE-PPO served as the baseline for RL-based fine-tuning. We ran the supervised-learning, ROUGE-PPO, and RewardModel-PPO for 2, 10, and 10 epochs, respectively, with a batch size of 16, 4, and 4. We selected the best final model based on the validation set.

Table 2 presents our model's performance. Compared with the zero-shot and ROUGE-PPO methods, our RM method achieved higher ROUGE and BLEU scores and generated slightly larger vocabularies. However, there is little difference between random sampling and active learning and the supervised learning method outperformed all fine-tuning methods, generating more reasonable summaries with more diverse vocabularies.

5 Discussion

In the context of small datasets, active learning showed better performance compared to random sampling. However, attaining high accuracy in large datasets necessitates numerous epochs and iterations within each query loop to construct an appropriate training model. This requirement incurs a substantial demand for computational resources and time. Moreover, a challenge that arises in the case of large datasets is that, for a fixed set of hyperparameters, the model may exhibit overfitting in small labelled data sizes and underfitting as the data size increases, making it challenging to select appropriate parameters. Unfortunately, the small-text library is not sufficiently adaptable to address this issue in large datasets.

In our end-to-end experiments, RL-based fine-tuning methods fail to exhibit a promising performance, with ROUGE-PPO even underperforming zero-shot. This could be attributed to the high-dimensional discrete action space of NLP tasks coupled with the sparsity of rewards, as previously noted by Choshen et al. (2019). Notably, Ramamurthy et al. (2022) suggests that the optimal approach would be to apply RL to a supervised-learning fine-tuned model. Nevertheless, the combination of reward models and RL demonstrates superior performance compared to PPO-ROUGE, indicating the potential benefits of aligning RL with human preferences. Despite of this, it is often difficult for RL to tune and may require significant resources to achieve optimal results. Active learning, however, does not appear to provide an advantage in the final end-to-end experiments, potentially due to its limitations in handling large data sizes, as previously discussed.

6 Conclusion

In this study, we proposed a framework, ARLHF, that combines active learning with reinforcement learning to fine-tune a large language model for text summarization tasks. Our approach demonstrated several advantages, such as reduced annotation costs, improved summarization performance, and generalizability to different tasks and domains.

Our end-to-end experiments on the widely-used CNN/Daily Mail dataset revealed that our proposed reinforcement learning method, when compared to zero-shot and ROUGE-PPO approaches, achieved higher ROUGE and BLEU scores. However, we observed little difference between random sampling and active learning, and the supervised learning method outperformed all other fine-tuning

metric	zero-shot	RM-random sampling	RM-active learning	supervised-learning	ROUGE-PPO
ROUGE1	0.354	0.357	0.356	0.389	0.339
ROUGE2	0.135	0.137	0.136	0.162	0.123
ROUGEL	0.233	0.235	0.234	0.260	0.224
BLEU	0.077	0.077	0.079	0.099	0.062
msttr-100	0.692	0.693	0.691	0.712	0.705
median_pred_length	55	55	57	72	47
min_pred_length	1	12	14	29	26
max_pred_length	94	93	94	94	92
vocab_size	37271	37798	37920	44558	35475

Table 2: Performance of 4 different fine-tuning methods

methods, generating more reasonable summaries with diverse vocabularies.

Moving forward, future work may focus on addressing the limitations of active learning in large datasets, refining the selection of hyperparameters, and exploring ways to enhance the adaptability of libraries for more efficient and accurate text summarization. Further research could also investigate the combination of active learning and reinforcement learning for other NLP tasks to establish the generalizability and robustness of the ARLHF framework.

References

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning, 2019. URL https://doi.org/10.48550/arXiv.1904.00370.

Tong Luo, Kurt Kramer, Dmitry B. Goldgof, Lawrence O. Hall, Scott Samson, Andrew Remsen, and Thomas Hopkins. Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research*, 6(20):589–613, 2005. URL http://jmlr.org/papers/v6/luo05a.html.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning, 2011. URL https://doi.org/10.48550/arXiv.1112.5745.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples, 2021. URL https://doi.org/10.48550/arXiv.2109.03764.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

Olivier Bachem, Mario Lucic, and Andreas Krause. Scalable k-means clustering via lightweight coresets. *KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, pages 1119–1127, July 2018. URL https://doi.org/10.1145/3219819.3219973.

Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning, 2019. URL https://openreview.net/forum?id=rJl-HsR9KX.

Cody Coleman, Edward Chou, Sean Culatana, Peter Bailis, Alexander C. Berg, Roshan Sumbaly, Matei Zaharia, and I. Zeki Yalniz. Similarity search for efficient active learning and search of rare concepts, 2021. URL https://openreview.net/forum?id=G67PtYbCImX.

Florian Schmidt. Generalization in generation: A closer look at exposure bias. *arXiv preprint arXiv:1910.00292*, 2019.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

Samuel Kiegeland and Julia Kreutzer. Revisiting the weaknesses of reinforcement learning for neural machine translation. *arXiv preprint arXiv:2106.08942*, 2021.

Abdelrhman Saleh, Natasha Jaques, Asma Ghandeharioun, Judy Shen, and Rosalind Picard. Hierarchical reinforcement learning for open-domain dialog. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8741–8748, 2020.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. On the weaknesses of reinforcement learning for neural machine translation. *arXiv preprint arXiv:1907.01752*, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, page 16857–16867, December 2020. URL https://dl.acm.org/doi/10.5555/3495724.3497138.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020. URL https://doi.org/10.48550/arXiv.1906.08237.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL https://doi.org/10.48550/arXiv.1908.10084.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. Efficient few-shot learning without prompts, 2022. URL https://doi.org/10.48550/arXiv.2209.11055.

Christopher Schröder, Lydia Müller, Andreas Niekler, and Martin Potthast. Small-text: Active learning for text classification in python, 2023. URL https://doi.org/10.48550/arXiv.2107.10314.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*, 2022.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21 (140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.