

Make The Most of Prior Data: A Solution for Interactive Text Summarization with Preference Feedback

Duy-Hung Nguyen^{1,*}, Nguyen Viet Dung Nghiem^{1,*}, Bao-Sinh Nguyen¹, Dung Tien Le¹, Shahab Sabahi¹, Minh-Tien Nguyen^{1,2}, Hung Le³

¹Cinnamon AI, 10th floor, Geleximco building, 36 Hoang Cau, Dong Da, Hanoi, Vietnam.
{hector, henry93, simon, nathan, sshahab, ryan.nguyen}@cinnamon.is

²Hung Yen University of Technology and Education, Hung Yen, Vietnam.
tienm@utehy.edu.vn

³Deakin University, Australia.
thai.le@deakin.edu.au

Abstract

For summarization, human preferences is critical to tame outputs of the summarizer in favor of human interests, as ground-truth summaries are scarce and ambiguous. Practical settings require dynamic exchanges between humans and AI agents wherein feedback is provided in an online manner, a few at a time. In this paper, we introduce a new framework to train summarization models with preference feedback interactively. By properly leveraging offline data and a novel reward model, we improve the performance regarding ROUGE scores and sample-efficiency. Our experiments on three various datasets confirm the benefit of the proposed framework in active, few-shot and online settings of preference learning.

1 Introduction

The advent of AI has changed business practices, though the human involvement is still important. The human roles in interaction with AI-powered machines have been evolving under the concept of *human-in-the-loop (HITL)* (Zanzotto, 2019). HITL allows humans to actively participate in supervising AI systems by approving, rejecting, or re-labeling current outputs, and providing expert-guided advices to the system. It will also act as the unique source of external knowledge from humans. By observing the outputs of AI systems, humans can hand-pick some potential outcomes and then feedback to the models for better performance.

In NLP, document summarization is considered as a subjective task (Stiennon et al., 2020). They argued that it would be hard to quantify what makes a “good summary” without the human judgment input. Collecting human feedback and evaluating the crafted summaries from documents for building the training datasets is time-consuming (Wu et al.,

2021) and labor-expensive. It is true particularly where the domain knowledge is required. Moreover, Uc-Cetina et al. (2021) argued the importance of the user’s intentions for modeling Natural Language Understanding with high performance.

Given these attributions, we are more interested in deploying an *interactive* HITL-based text summarization framework, which continuously collects the user-feedback to consequently improve model prediction robustness. Here, the user’s intention is implicitly acknowledged as a factor influencing the extraction of important sentences from the source documents. Upon this formulation, the AI model will be trained with human-produced summaries and adapted as more human-feedback is fed in.

Previous studies used human feedback to rank the label of objects (Wirth et al., 2017), which employed reinforcement learning to minimize the user’s effort to provide feedback for training a ranker. However, these approaches rank the entire solution space including relevant and irrelevant pairs, which are a waste of computing power. To tackle this shortcoming, Siddhant and Lipton (2018) employed Bayesian Optimization, which substitutes the standard uncertainty-based acquisition functions for active learning. However, the model still consumes a lot of computing power for a larger number of iterations, and is vulnerable to the curse of dimensionality of input data. Recently, researchers proposed to learn a reward model simulating human preferences (Ziegler et al., 2019). The reward model was then used to translate real-time human feedback to the reward score for fine-tuning the model under RL training. The method was designed for online learning of language models and required numerous interactions to achieve good performance. However, how to achieve sample-efficiency is still an open question.

In this paper, we propose a novel interactive pref-

*These authors contributed equally to this work.

erence learning for the summarization task. To do that, we fine-tune a pretrained extractive summarizer as the backbone with reinforcement learning by using a reward model that enforces the distance-based order of preferences. The reward model is trained to differentiate two summaries regarding the topic, length, and quality (human preferences). To enable sample-efficiency, we propose to utilize *offline* data, which was previously used to pretrain the model. We show that naively using the offline data is harmful for preference learning. Instead, we introduce two mechanisms to selectively sample offline data in favor of human feedback learning. Our sampling strategies focus on low-rewarded samples or documents which are similar to fine-tuning data. We demonstrate that our method can be used in various settings: active, few-shot and online learning. Tested on three summarization datasets, our method consistently achieves significantly better results compared to competitive baselines in each setting. In summary, our contribution is three-fold:

- We propose a new RL-based preference learning system for the summarization task by using a novel reward model.
- We propose sampling mechanisms to efficiently leverage offline data for preference learning of extractive summarization.
- We conduct extensive empirical studies on three summarization datasets, showing that our proposed method outperforms competitive baselines in various settings such as active, online, and few-shot learning scenarios.

2 Related Work

2.1 Summarization with RL

Direct reward The most direct way to have a reward function in reinforcement learning for summarization is to match the candidate summary to the reference (gold) summary (Narayan et al., 2018; Paulus et al., 2018). Put in the HITL setting, humans are required to provide the gold summary for training the machine, which is prohibitively expensive and can be ambiguous. Thus, we focus on rewards constructed from preference feedback, in which humans only need to indicate the better summary between two candidate summaries.

Preference reward As stated in the introduction, the preference from humans is much more accessible and consistent. In this approach, the frame-

works (Gao et al., 2018; Stiennon et al., 2020; Nguyen et al., 2021) consist of two main steps: 1) Preference learning that gives a score, which mimics human evaluation, to a summary of a document. 2) Reinforcement learning based on the reward model. These works have not examined the summarization problem in an interactive training scheme, which will be addressed in this paper.

2.2 Preference learning in NLP

Preference learning aims at obtaining the ranking (i.e. total ordering) of objects from pairwise preferences, in which the linear Bradley-Terry (BT) model (Bradley and Terry, 1952) is one of the most studied methods. Later, the APRIL framework (Gao et al., 2018) shows that it can reduce the number of required comparisons by using active selection with uncertainty sampling (P.V.S and Meyer, 2017). On the other hand, the OpenAI framework (Stiennon et al., 2020) uses a neural structure to predict the score given a document and its summary. Its advantage is the learning with direct human evaluation (score for each summary) or human preferences (comparison between 2 summaries). However, the input space is the product of two documents, so it needs much more human feedback to achieve good performance. In contrast, our proposed reward model treats the input as only one document, and thus is simpler and has fewer parameters in the learning process. That is more suitable for the interactive learning context where the number of training samples is limited.

Reinforcement learning is a popular and effective approach to utilize human preferences. The APRIL framework (Gao et al., 2018) trains a summarizer by using a pretrained preference model. The work of Ziegler et al. (2019) employs preference-based RL to fine-tune a deep language model for sentiment classification and document summarization tasks. They only rely on online data during fine-tuning. We instead make use of offline data to accelerate the learning process.

3 Background

3.1 The backbone model

We use BERTSUM (Liu and Lapata, 2019) as the backbone for extractive summarization. Given a document with a set of sentences, the model uses BERT (Devlin et al., 2019) for learning hidden vectors of sentences by using the modified [CLS] token of each sentence. The vectors are fed into

an inter-sentence layer by using the Transformer for learning the inter-relationship among sentences. Important sentences are extracted by using a sigmoid function for sentence importance estimation.

3.2 Fine-tuning with reinforcement learning

Proximal policy optimization (PPO) To fine-tune the backbone with interactive feedback, we treat the summarization process as a sequential decision-making process so that we can employ RL (Stiennon et al., 2020; Ziegler et al., 2019). The RL agent traverses all sentences in the original document, and at each time step, it classifies the current sentence s_i into two labels: important ($y_i = 1$) and unimportant ($y_i = 0$). We consider the backbone model trained with supervised learning as the initial policy. The policy is then optimized by using PPO (Schulman et al., 2017) as our RL method. The objective of training PPO is:

$$J^{PPO}(\theta) = \mathbb{E}[\min(ra(\theta)\hat{A}_{\theta_{old}}(s_i, y_i), \text{clip}(ra(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_{\theta_{old}}(s_i, y_i))]$$

where θ is the current policy’s parameters, $\hat{A}_{\theta_{old}}(s_i, y_i)$ is the advantage calculated at the old policy parameters θ_{old} before each updated policy iteration, by using any advantage estimation algorithm to transform the rewards (Schulman et al., 2016), and $ra(\theta) = \frac{\pi_{\theta}(y_i|s_i)}{\pi_{\theta_{old}}(y_i|s_i)}$ is the ratio between the new policy and the old policy. This is the idea of importance sampling that evaluates the new policy with samples collected from the older policy. If the ratio ra falls outside the range $1 - \epsilon$ and $1 + \epsilon$, the advantage function will be clipped. PPO uses the objective to avoid big changes between new and old policies.

Reward schemes Besides the final reward that evaluates the quality of the whole summary, we follow Pasunuru and Bansal (2018); Li et al. (2019) to use an additional reward to constrain policy updates. Let π_{θ_0} denotes the supervised trained backbone model and π_{θ} is the one that we optimize with RL. The reward at the time step i is:

$$r_i = -\beta_{KL} \log \left[\frac{\pi_{\theta}(y_i|s_i, D)}{\pi_{\theta_0}(y_i|s_i, D)} \right] + \mathbb{1} \times (i = n) \times r^M$$

where β_{KL} is the KL coefficient, n is the final time step corresponding to the total number of sentences in the document D . For the intermediate time step i ($i < n$), the reward is the negative KL

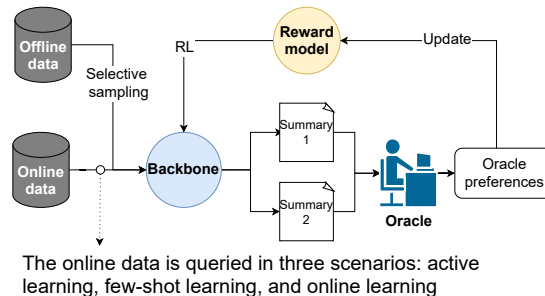


Figure 1: The overview of our framework. In which the backbone is in charge of generating two summaries for a document. Then the oracle selects which summary is better for a given document. The reward model afterward transforms the oracle’s preference into a discrete signal to optimize the backbone. Our framework contains two novel components: efficient sampling from offline data and the preference-guided reward model.

divergence between the output distribution of the backbone model and the current policy. For the final time step i ($i = n$), when the model obtains the complete summary S , the reward model takes the summary and the original document to produce the final reward r^M (see Section 5).

4 Interactive Learning with Online RL

4.1 Problem formulation

In our problem, a backbone summarizer and a reward model are pretrained with *offline* data by using supervised learning. We aim to update the models to adapt with a stream of *online* data by using RL. Given a document D from the online data, the backbone (Section 3.1) extracts two summaries for an oracle (i.e., a human) to provide preference feedback. Each time the backbone receives feedback in one interaction. After receiving feedback from the oracle, the reward model is fine-tuned by using supervised learning (Section 5). The feedback is then translated to a reward via a reward model, which is later used to fine-tune the backbone by using RL (Section 3.2), thereby improving the summary quality for future interactions.

This interactive mechanism is applicable to three scenarios: active, online, and few-shot learning (Section 4.2). Offline data can be employed during online fine-tuning to speed up the learning process by using the selective sampling method (Section 4.3). The common objective of all scenarios is to maximize summarization quality while minimizing the number of interactions. We describe the overview of our framework in Figure 1.

4.2 Interaction plots

Active learning Given a pool of unlabeled samples, the agent queries the most informative samples at each iteration to create summaries. Then the agent asks the oracle to select which summary is better for the queried documents. The agent learns from the feedback of the oracle subsequently. This setting has been studied by Gao et al. (2018).

Online learning Different from active learning, for online learning, each unlabeled instance is typically drawn one at a time from the data stream, continuously being sent to the system. The agent processes the data and is not allowed to choose the document. Ziegler et al. (2019) is the early work that promotes the online learning scenario.

Few-shot learning Similar to the online setting, the system passively receives samples from the data stream. This time, it is only given a few unlabeled samples (up to 4 documents) and manages to learn from the data by continuously producing summaries and receiving feedback. To our best knowledge, this is the first time that the few-shot setting is examined in preference learning.

4.3 Efficient interaction from offline data

To achieve sample efficiency in terms of the number of interactions between the backbone and the oracle, we argue that it is critical to leverage prior data used in the pretraining phase. We observe that the naive use of prior data by using random sampling is inefficient. Hence, we propose two novel offline-data sampling methods: low-reward and document-similarity. We describe all sampling mechanisms as follows.

Random sampling (Random) We randomly select k offline documents and combine them with online data to fine-tune the backbone model.

Low-reward sampling (LRS) Intuitively, high-quality summaries should be given a higher reward by the reward model. Samples that have extracted summaries with low rewards have not been well-learned by the summarizer. As the result, we select top k documents having the lowest rewards from the reward model.

Document-similarity sampling (DSS) Conventionally, the backbone outputs good summaries when the training and testing distributions are similar. Therefore, we sample offline documents that share similar semantics with online documents. To

do that, we encode the documents by using BERT (Devlin et al., 2019), and then compute their similarity by using the Cosine distance. k offline documents have the lowest distance to the online counterparts are selected for online training.

5 The Reward Model for Preference Learning

5.1 Training procedure

We construct a reward model to generate preference rewards for finetuning the backbone summarizer with RL. The reward model should simulate human preferences, capable of assigning a higher reward to the preferred summary. For preference learning, we use the relative order between two summary candidates to train the reward model. The correct order is given by humans via their preferences.

For traditional learning, the model has access to training documents of the dataset. Each document D has a gold summary S_g , which is assumed to be preferred to other candidate summaries (silver or machine-generated S_m), denoted by $S_g \succ S_m$. The training data is in the form of a triplet (D, S_g, S_m) where $S_g \succ S_m$. We argue that this is insufficient to achieve a good reward model. From our observation, before reaching the quality of gold summaries, the machine often produces off-topic and too-short answers. In addition, the work of Maxwell et al. (2017) also suggested that the oracle prefers longer summaries because they feel that longer summaries are more readable, clear, and informative. Thus, we introduce three objectives as follows.

The topic objective Aiming to detect the document’s topic: (D, S_g, S_m) . We generate the training set of the **topic objective** as follows. For each document D that has a gold summary S_g , we select a summary S'_g from a different document D' (randomly chosen) and expect that $S_g \succ S'_g$. If we miss the human summary, we could use machine-generated summaries instead $((D, S_m, S'_m)$ such that $S_m \succ S'_m$).

The length objective Aiming to detect the amount of summary information (i.e. the length of the summary). For each document D , we used a pretrained summarization model to generate a long (S_{ml}) and a short (S_{ms}) summary to obtain the triplet (D, S_{ml}, S_{ms}) . The correct order is $S_{ml} \succ S_{ms}$.

The quality objective Aiming to detect the general quality of a summary. For each document D , we generate a summary S_m which has a similar length to the human summary and include the gold summary to compose the triplet for training (D, S_g, S_m) . We note that the data of this objective is similar to the traditional data used in [Stiennon et al. \(2020\)](#), which needs humans to provide a gold-silver summary or correct order between arbitrary summary candidates. On the contrary, the data of **topic** and **length** objectives can be created programmatically and thus save human effort.

5.2 Respective-order mapping

For summarization, we argue that the reward model’s interpretation of ordering should be determined based on the distance between the summary and document representations. Therefore, we aim to learn global representing mapping ϕ such that:

$$S_1 \succ S_2 \iff d(\phi(D), \phi(S_1)) < d(\phi(D), \phi(S_2))$$

where $d()$ is the Euclidean distance, $\phi(D)$ is the representation of the document D in the respective-order mapping. Then, the reward for a summary S is computed as follows.

$$r^M(D, S) = \frac{\text{score}(\phi(D), \phi(S)) - \text{score}_{\min}}{\text{score}_{\max} - \text{score}_{\min}}$$

where $\text{score}(D, S) = \frac{1}{1 + \exp(d(\phi(D), \phi(S)))}$ is the unnormalized score between D and S ; score_{\min} and score_{\max} are the minimum and maximum score over all pairs of document-summary, respectively. We name our reward model as ROMSR (Respective-Order Mapping Score Reward).

5.3 Representation learning

To train ϕ , first, we embed all documents that appear in triplet sets. To obtain important features of a document, we use the joint-embedding of a Transformer (capture a general meaning of the document) ([Reimers and Gurevych, 2019](#); [Devlin et al., 2019](#)) and keyword feature vectors (capture important keywords in the document) ([Sharma and Li, 2019](#)). We denote this transformation as $f(D)$.

In fact, this feature vector $f(D)$ can be used to compare the similarity between two documents. However, it is not robust enough to satisfy the relative comparison from the triplets. Therefore, we learn a new embedding ($\text{encode}(f(D))$) that tries to follow all the triplet conditions. In order to be a good representation of $f(D)$, the embedding is

also reinforced by the reconstruction loss of the autoencoder. The embedded representation is our target mapping $\phi(D) = \text{encode}(f(D))$. The loss for the learning of the document representation \mathcal{L} is the combination of the autoencoder reconstruction loss \mathcal{L}_{AE} and the respective-order loss \mathcal{L}_{RO} (the triplet loss) as follows.

$$\mathcal{L} = \mathcal{L}_{AE} + \mathcal{L}_{RO}$$

$$\mathcal{L}_{AE} = \sum_{D \in \mathbb{D}_{US}} \|f(D) - \text{decode}(\phi(D))\|$$

$$\mathcal{L}_{RO} = \sum_{(D, S, S') \in T} \max(0, d(\phi(D), \phi(S)) - d(\phi(D), \phi(S')) + \alpha)$$

where \mathbb{D}, \mathbb{S} is the set of original/summarized documents, T is the union of all triplet sets, and α is the margin hyperparameter controlling the stretch in the representation space ([Yu et al., 2018](#)).

6 Experimental Setup

6.1 Datasets

We use three benchmark summarization datasets for our evaluation. **BillSum** comprises 22,218 US bill and human-written summary pairs collected from the US Government Publishing Office ([Kornilova and Eidelman, 2019](#)). The data is split into 18,949 training and 3,269 testing bills. **Reddit TIFU** is an English dataset collected from Reddit. It contains 12,000 posts divided into TIFU-long and TIFU-short (documents have less than 400 words) ([Kodaira and Komachi, 2018](#)). **Livedoor** contains Japanese articles crawled from the Livedoor News website. Each article consists of three summary sentences written by editors ([Kim et al., 2019](#)).

Our experiments use the training set to pretrain the backbone and reward models. Then, we randomly select 5000 samples from the original training data to create offline data. The online data is created from the original validation data. For active learning and online learning settings, we sample up to 320 documents from the online data. For few-shot learning, only 4 documents are randomly selected as online data. We aim to minimize the number of interactions to reach the highest performance on the whole online dataset.

6.2 Simulated interactions

Given one original document with two corresponding generated summaries and a standard metric ROUGE score, the oracle knows the ground-truth summary, and prefers the summary that has the higher ROUGE score w.r.t the ground-truth. Theoretically, the ideal data for preference-based interactive learning is consistent preferences in which a higher-scored summary is always selected. However, we believe that perfect selection is impossible for real-world applications because humans can occasionally misinterpret the intention when the presented candidates have similar qualities. Therefore, in this work, we consider noisy preferences with uniform probability $nc \in [0, 1]$, that randomly selects which summary is better.

6.3 Baselines

We construct a standard model (**baseline**) that just finetunes the backbone summarizer (see Section 3.1) with online data, coupled with our proposed reward model and PPO training. The combination of BERT and PPO is shown effectively in the offline document summarization task (Nguyen et al., 2021). The baseline is similar to that of Ziegler et al. (2019), which uses human preferences and PPO to finetune language models. However, the latter uses a different reward model.

To show the efficacy of our reward model, we compare ours with the reward model of Ziegler et al. (2019) (**OpenAI**) and the uncertainty sampling reward model in **APRIL** (Gao et al., 2018) in the preference prediction task (see Section 7.1).

For few-shot and online learning, we compare our methods with the standard baseline. For active learning, we also compare our methods with **APRIL**. For all settings, we build a baseline that randomly samples from offline data to compare against our proposed sampling techniques: **LRS** and **DSS**. We note that all models share the same backbone and RL training. The baseline, Random, LRS, and DSS share the same reward model and only differ in the sampling techniques.

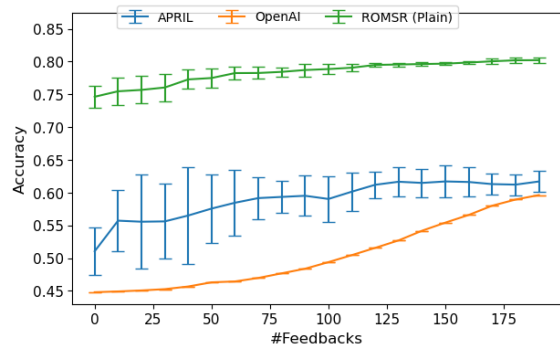
6.4 Evaluation metrics

We use ROUGE-scores (Lin, 2004) for our evaluation, in which ROUGE-1 is the representative score. ROUGE-2 and ROUGE-L are reported in the Appendix. We also report the number of interactions to show that our novel sampling techniques can significantly speed up the agent’s learning.

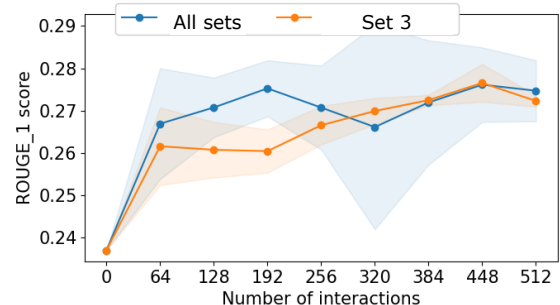
7 Results and Discussion

7.1 Reward model study

We first observe the efficacy of the proposed reward model in the summarization process. To do that, we compare our reward model to the reward models of **APRIL** (Gao et al., 2018) and **OpenAI** Ziegler et al. (2019). For the preference prediction task, the reward scores assigned to candidate summaries are used to determine the preferred summary. In general, the preferred summary is the one with a higher score. We select a subset of documents from the datasets to train the reward model. Each document can be used to construct three triples corresponding to the three objectives mentioned above. The number of pretraining/interactive training/testing documents is 1000/1000/1000 for Billsum and Livedoor, and 2000/2000/2000 for Reddit TIFU, respectively.



(a) Average testing accuracy of APRIL, OpenAI and ROMSR when using all three sets.



(b) Active learning on RedditTIFU dataset of ROMSR when using all three sets and set 3 only

Figure 2: Reward model: (a) Accuracy of 3 models on all sets (b) The ROUGE-1 score of our ROMSR.

Figure 2 (a) demonstrates that our reward model (ROMSR) significantly outperforms other methods when training on all three objectives. **APRIL** uses a linear regression model combined with a heuristics function that is tuned for the quality objective. This specific design limits the model to handle all three objectives. **OpenAI**’s results exhibit slow learning progress because it is a large neural network using

joint features from a pair of texts, while our reward model learns to use respective-order on a feature space of a single document. Moreover, the reconstruction loss from the autoencoder helps the model to avoid overfitting to small training samples.

To verify the necessity of using three objectives for training the reward model, we compare two versions of our reward model: one is trained with all sets of objectives and another with the third objective (the traditional quality objective). Two reward models are used in active learning to finetune the backbone. Figure 2(b) shows that training on all three objectives converges faster.

7.2 Fine-tuning backbone model using human preferences

This section shows our comprehensive experiments on three datasets and three scenarios to prove the effectiveness of our human-preference-guided learning with RL, and our LRS and DSS sampling.

Active learning We compare our sampling techniques with the baselines, including random offline sampling and methods without offline sampling, such as the standard baseline and APRIL. Figure 3 shows that combining online data with randomly selected samples harms the agent during online fine-tuning. Meanwhile, selecting prior low-reward samples and similar documents boosts accuracy faster than not using offline data. After 64 interactions, ROUGE-1 reaches almost the highest. To validate the actual impact of ROMSR, we also test the reward model in the RL pipeline compared to the APRIL’s reward. Due to the OpenAI reward is inferior to the APRIL’s reward, we ignore the OpenAI’s reward in this experiment. The results in Figure 3 show that APRIL (orange) is inferior to our ROMSR (blue) in all datasets.

Few-shot learning Figure 4 describes the quality of the models: LRS, DSS, Random, and the standard baseline during interactions. The standard baseline without offline sampling can not improve the summarization agent with a few interactions and eventually worsen the agent. In contrast, LRS and DSS can significantly improve the model after 4 interactions. Naively using the offline data still harms the model, similar to active learning.

Online learning Due to computation limitations, we only examine this setting on the Reddit TIFU dataset. Also, prior experiments demonstrated that LRS and DSS are equally good sampling strategies.

As DSS runs much faster than LRS, we choose DSS as our representative method in this experiment. We report the performance of DSS, Random, and the standard baseline in Figure 5.

The results show that random sampling is again inferior to other methods in this online scenario. The standard baseline shows good performance yet becomes unstable and drops performance later. Our DSS demonstrates fast and stable convergence, consistently achieving the highest ROUGE-1 score throughout interactions.

7.3 Ablation study and model analysis

Impact of top- k selected offline documents To investigate the effect of k -the number of sampled offline documents, we try different k values ($k = 1$ and $k = 4$) on the RedditTIFU dataset, and the results are depicted in Figure 6a. As observed, $k = 4$ is slightly better, but $k = 1$ runs and converges faster. Hence, we use $k = 1$ in all experiments.

Random sampling makes low rewards We investigate the failure of random sampling. We keep track of the reward assigned to randomly selected offline documents across interactions. Figure 6b shows that, in general, DSS and LRS give better samples, indicated by higher assigned rewards, than random sampling. DSS and LRS’s rewards are also much more consistent, showing less variance than random sampling. Thus, offline documents from random sampling provide little benefit and make the training unstable and unreliable.

We also analyze the distribution characteristics of selected offline documents in Figure 6c. The green line represents the distribution of similarity scores between online and random offline samples. The red one is the distribution of similarity scores between online and DSS samples. It is noticeable that the similarity score of DSS is in the range from 0.0 to 0.15, whereas this score of Random is up to 0.5. It is expected that sampling offline data will create a coherent training distribution for online learning, but Random skews the training distribution. Therefore, Random harms the performance.

Running time We assess the running time of our proposed method compared to the baseline on Reddit TIFU and BillsSum in Table 1. The assessment is conducted on a single Tesla T4. On Reddit TIFU, our DSS takes nearly 3 hours (64 interactions) to reach almost the highest ROUGE-1 score of 28.6, while the baseline reaches only 26.0 of ROUGE-1 after 64 interactions. The baseline takes only

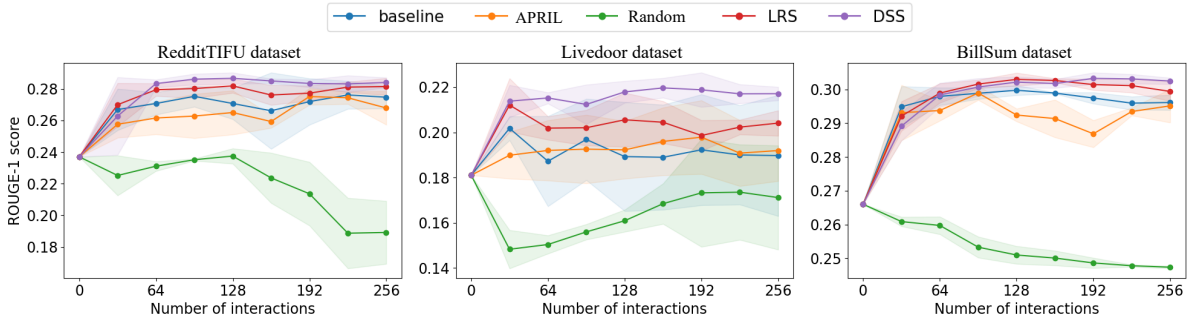


Figure 3: Active learning: ROUGE-1 with the mean and standard deviation over 5 runs.

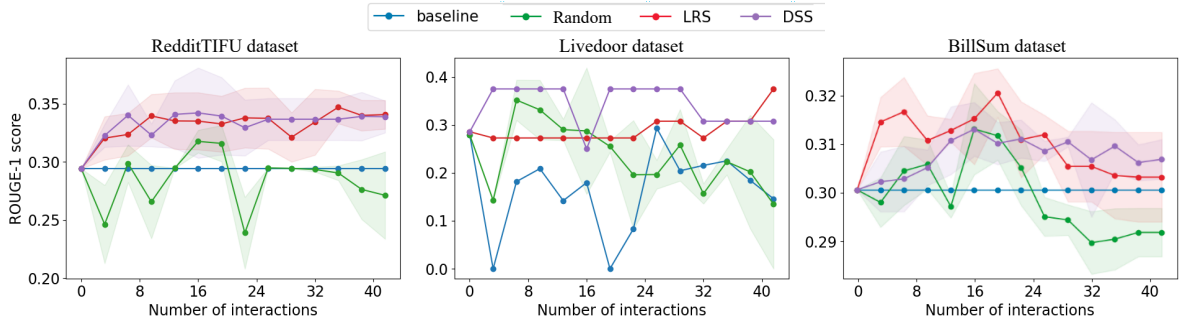


Figure 4: Fewshot learning: ROUGE-1 with the mean and standard deviation over 5 runs.

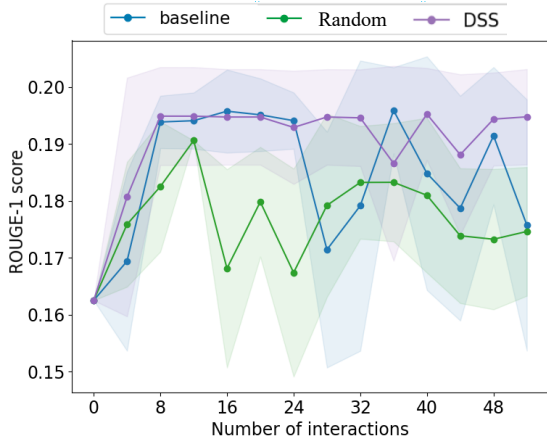


Figure 5: Online learning on Reddit TIFU: ROUGE-1 with the mean and standard deviation over 5 runs.

30m faster than the DSS. In light of BillsSum, the DSS is slower than the baseline, about 50 minutes for 96 interactions. However, the DSS is better than the baseline in terms of ROUGE-1. Despite that our proposed method takes more computing resources than the baseline, our approach requires much fewer interactions, saving human resources in the real HITL setting. In terms of real-time interactive feedback, it takes around 3 minutes for training in each interaction on the Reddit dataset, and approximately 1 minute for training in each

Method	Interactions	Run time	ROUGE-1
The Reddit TIFU dataset			
Baseline	64	2h20m	26.0
DSS	64	2h50m	28.6
The BillsSum dataset			
Baseline	96	50m	29.4
DSS	96	1h40m	30.3

Table 1: The running time of active learning with offline sampling.

interaction in the BillsSum dataset, which is not really real-time. It is noted that the amount of time to provide feedback is not included because we use simulated oracles. Hence, in actual applications, we can fine-tune the model while the oracle gives feedback because it takes time for the oracle to read the summary and decide their preference. We believe such a parallel process will make the methods feasible to use in real-time settings.

8 Conclusion

This work proposes a novel approach for learning reward functions in preference learning. By utilizing offline data with a reward model that focuses on the distance between the summary and the document, our method improves the ROUGE-scores

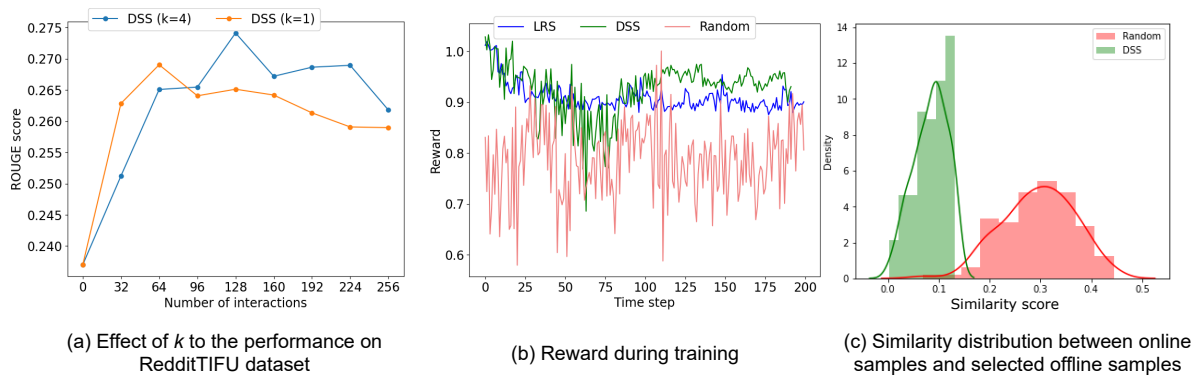


Figure 6: Ablation study: (a) Performance with different k values; (b) Quality of selected samples; (c) Semantic similarity between online documents and offline documents.

by 2 – 5% in comparison to the APRIL framework, while beating the random baseline by a large margin on three different datasets. Our experimental results also suggest that by applying low-reward sampling or document-similarity sampling, we can achieve efficiency in terms of both running time and the number of human interactions. Regarding limitations, our method is not tested with large online stream data, which may cause catastrophic forgetting. Future work will confirm our model’s effectiveness for abstractive summarization.

Acknowledgement

We would like to thank anonymous ACL ARR reviewers and Senior Area Chairs who gave constructive comments for our paper.

References

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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. Association for Computational Linguistics.

Yang Gao, Christian M. Meyer, and Iryna Gurevych. 2018. APRIL: Interactively learning to summarise by combining active preference learning and reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4120–4130, Brussels, Belgium. Association for Computational Linguistics.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of Reddit posts with multi-level memory networks. 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 2519–2531, Minneapolis, Minnesota. Association for Computational Linguistics.

Tomonori Kodaira and Mamoru Komachi. 2018. The rule of three: Abstractive text summarization in three bullet points. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56, Hong Kong, China. Association for Computational Linguistics.

Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. 2019. Deep reinforcement learning with distributional semantic rewards for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6038–6044, Hong Kong, China. Association for Computational Linguistics.

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

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

- David Maxwell, Leif Azzopardi, and Yashar Moshfeghi. 2017. A study of snippet length and informativeness: Behaviour, performance and user experience. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 135–144.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. [Ranking sentences for extractive summarization with reinforcement learning](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.
- Duy-Hung Nguyen, Bao-Sinh Nguyen, Nguyen Viet Dung Nghiem, Dung Tien Le, Mim Amina Khatun, Minh-Tien Nguyen, and Hung Le. 2021. [Robust deep reinforcement learning for extractive legal summarization](#). In *International Conference on Neural Information Processing*, pages 597–604. Springer.
- Ramakanth Pasunuru and Mohit Bansal. 2018. [Multi-reward reinforced summarization with saliency and entailment](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 646–653, New Orleans, Louisiana. Association for Computational Linguistics.
- Romain Paulus, Caiming Xiong, and Richard Socher. 2018. [A deep reinforced model for abstractive summarization](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- Avinesh P.V.S and Christian M. Meyer. 2017. [Joint optimization of user-desired content in multi-document summaries by learning from user feedback](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1353–1363, Vancouver, Canada. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. [High-dimensional continuous control using generalized advantage estimation](#). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *ArXiv preprint*, abs/1707.06347.
- Prafull Sharma and Yingbo Li. 2019. Self-supervised contextual keyword and keyphrase retrieval with self-labelling.
- Aditya Siddhant and Zachary C. Lipton. 2018. [Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909, Brussels, Belgium. Association for Computational Linguistics.
- Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. [Learning to summarize from human feedback](#). *ArXiv preprint*, abs/2009.01325.
- Victor Uc-Cetina, Nicolas Navarro-Guerrero, Anabel Martin-Gonzalez, Cornelius Weber, and Stefan Wermter. 2021. [Survey on reinforcement learning for language processing](#).
- Christian Wirth, Riad Akrouf, Gerhard Neumann, and Johannes Fürnkranz. 2017. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136).
- Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2021. [A survey of human-in-the-loop for machine learning](#).
- Baosheng Yu, Tongliang Liu, Mingming Gong, Changxing Ding, and Dacheng Tao. 2018. Correcting the triplet selection bias for triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 71–87.
- Fabio Massimo Zanzotto. 2019. Human-in-the-loop artificial intelligence. *Journal of Artificial Intelligence Research*, 64:243–252.
- Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. [Fine-tuning language models from human preferences](#). *ArXiv preprint*, abs/1909.08593.

A Hyper-parameters and Implementation details

Due to the limitation of resources, our backbone bases on BERT-base with 12 Transformer blocks with the hidden size is 768, and 110M parameters. We train our model with the learning rate of $1e-5$. During online fine-tuning with RL, we set k equals the number of online documents, where k is the number of selected offline data. The noisy preference probability nc is 0.1. For evaluation, we use ROUGE-score with parameters $-c 95 - m - r 1000 - n 2$.

B Additional Results

In Figure 8, we show ROUGE-2 and ROUGE-L scores with the same setting as section 7.2. The results of few-shot and online learning setting is depicted in Figure 9 and 7 respectively.

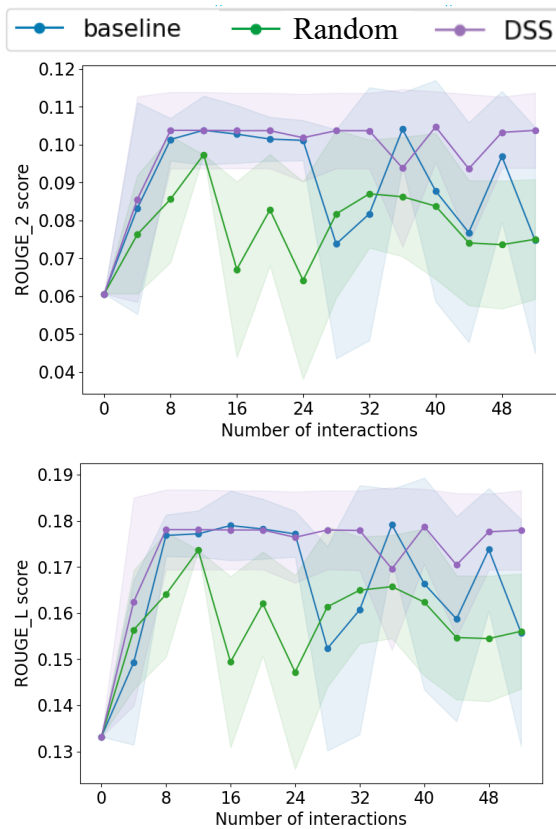


Figure 7: Online learning on RedditTIFU dataset: ROUGE-2 and ROUGE-L score with the mean and std over 5 runs

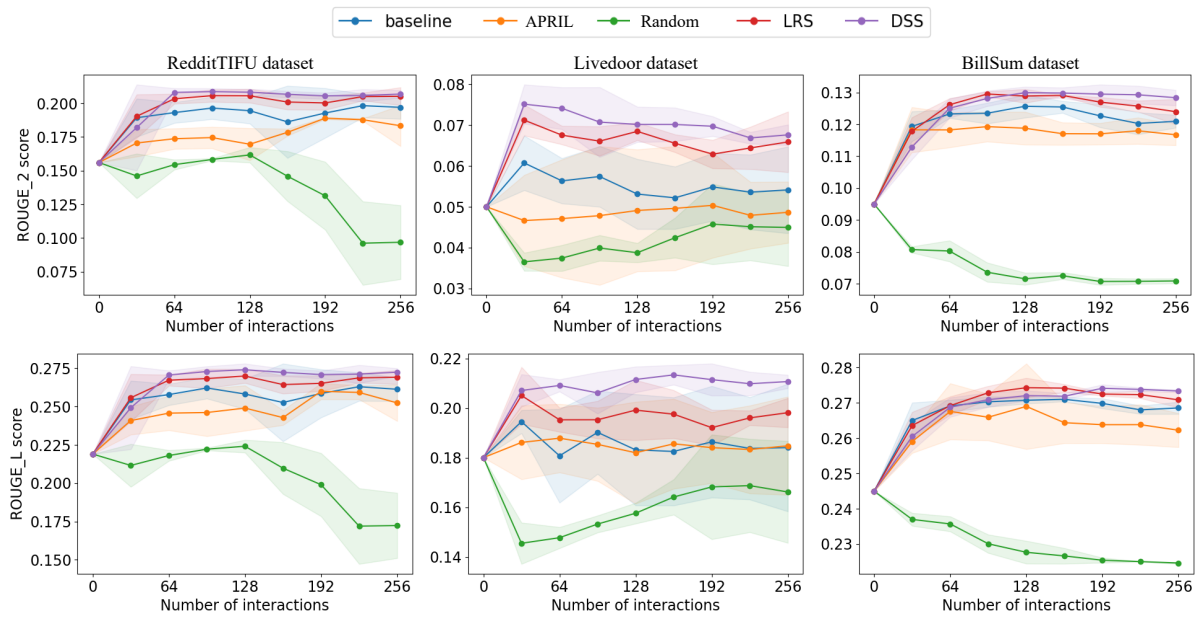


Figure 8: Active learning: ROUGE-2 and ROUGE-L score with the mean and std over 5 runs

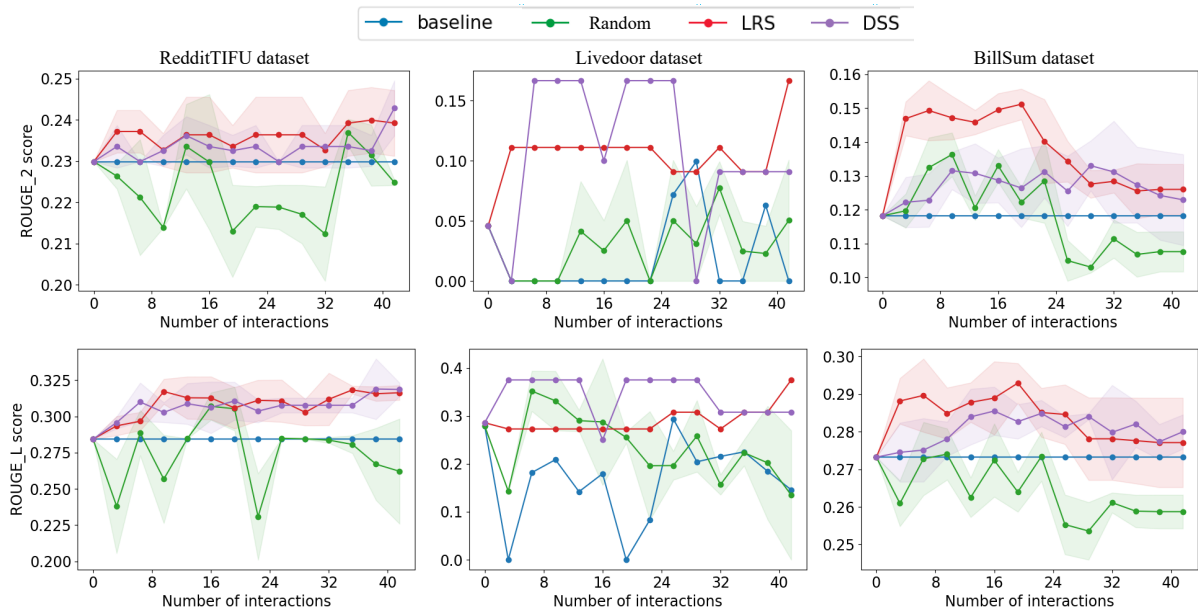


Figure 9: Fewshot learning: ROUGE-2 and ROUGE-L score with the mean and std over 5 runs