Human-in-the-loop training for language models

Haoyang Chen

Abstract

Machine learning has emerged as a cutting-edge technique for various tasks such as computer vision, natural language processing, and speech processing. However, due to the technique challenges and theoretical limitations associated with machine learning, these models are not always intelligent enough as human expect. This fact suggests that incorporating user knowledge into the system can be advantageous. Combining human domain knowledge and human feedback to a machine learning (ML) model not only promotes the performance of the system but also reduce the need for a large volume of construct training data. Human-in-the-loop is a field of research that is becoming increasingly important in the future since machine learning knowledge is still behind human domain knowledge in most realistic tasks, especially for those require creativity. The objective of human-in-the-loop is to train a more intelligent model by combining human knowledge and experience, improving the alignment between model performance and human behaviors. This paper is aimed at introducing the concept of human machine interaction from the perspective of machine learning, particularly focusing on the most popular technique that improving the model performance by human-in-the-loop training. Then the paper puts the spotlight on the application of human-in-the-loop training for natural language processing (NLP) context. Additionally, the paper explores the future research direction of human-in-the-loop under the context of NLP community.

1. Introduction

Deep learning is an area of artificial intelligence that achieving the goal of bridging the gap between automation of machine system and human intelligence. The successful applications of deep learning models cut across the boundary of many different fields, in which natural language processing and computer vision attracted most of the spotlights [1]. The success of deep learning can be attributed to its larger models, which can consist of hundreds of millions of parameters. This vast number of parameters provides the model with greater degrees of descriptive capability, allowing it to performs better in awe-inspiring description tasks [2].

Admittedly, increasing the model size, which corresponding to the number of parameters, can significantly improve the descriptive capability of a model. The volume of training data with labels that needs from the model, however, will also become massive as the increment of the model parameters. One of the challenges in the deep learning research especially in NLP community is the demand for structured training data often outpaces the available supply, which can limit the accuracy and effectiveness of deep learning models. This is mainly because data annotating and labeling is a labor-intensive and time consuming work [3], and thus also expensive.

To tackle this problem, some researchers proposed to build new dataset by data augmentation techniques or generating shadow dataset [4-6]. Even though data augmentation is a solution to solve the problem of dataset resource for large scale model, it is not the best solution to solve the problem fundamentally. Large scale machine learning models are still data hungry. In addition, this branch of techniques is not only encouraging an idea that “A large dataset is All you need”, which is unsustainable, but also showing there is still a deep gap between the current artificial intelligence system and human intelligence. As we know, humans are capable of learning new tasks rapidly even with few learning samples and prior experience. For example, a child is able to identify the same person from a large number of photos after given a few photos of a stranger [7]. Another branch of techniques to solve this challenge is transfer learning and pretrain-finetune framework [8], which have achieved a lot of incredible results, particularly in NLP. Based on this concept, a further development led to a new and young research area, which is Few-Shot learning [7, 9]. Human-in-the-loop training or human-machine interactive learning is research topic that being explored under this context.

Since this paper is aimed at introducing the human-in-the-loop training or human-machine interactive learning under the background of NLP community, we will only focus our discussion on a sub-topic that building better language model with human feedback. Though, it is still worth mentioning that when talking about human-in-the-loop machine learning, the circle of this research topic is much larger than the point that this paper indicates.

1.1Human-in-the-loop conception

"Human-in-the-loop" (HITL) is referring to a human's involvement or participation in a process or system that involves automated or machine-based decision making. It means that a human is still involved in the decision-making process, providing oversight, feedback, or guidance to ensure that the automated or machine-based decision-making process is effective, ethical, and aligned with human goals and values. This conception, is in line with the core idea of Human-Computer Interaction (HCI), emphasizes the interaction between human and machine.

Even though the human-in-the-loop conception is a wide range research topic that falls on the intersection of computer science, cognitive science, and psychology, most recent researchers render this approach in areas such as artificial intelligence, machine learning, and robotics, where there is a need to balance automation and human expertise. These works [10-13] follow the same idea that integrate human knowledge and wisdom to an artificial intelligent system to boost the system performance.

Evidently, as shown in Figure 1, the combination of human-in-the-loop and machine learning is becoming the new darling, no matter in academia or industry [3].

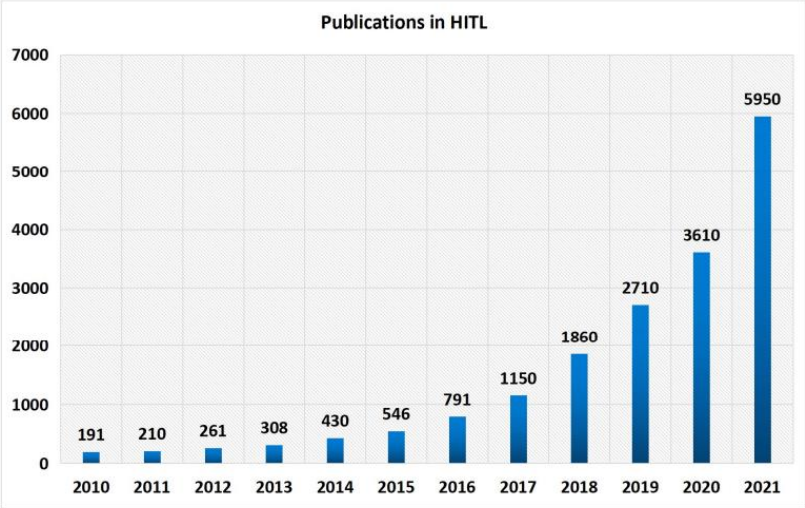


Figure 1. The search volume of the key words “human-in-the-loop” and “machine learning” is dramatically increasing in the recent years.

1.2Natural language processing

NLP has become an increasingly important area of research in recent years, due to the explosion of digital data and the need to understand and analyze large volumes of text. It is a multidisciplinary field that combines techniques from linguistics, computer science, and artificial intelligence to enable computers to process, analyze, and generate human language [14].

NLP has a wide range of applications, including machine translation, sentiment analysis, question-answering systems, chatbots, and text-to-speech systems. For example, machine translation systems use NLP techniques to translate text from one language to another, while chatbots use NLP to enable natural language conversations between humans and machines.

From statistical languages modeling to neural languages modeling, the NLP community experienced a series of profound evolution during the recent years. A lot of previous state-of-the-art techniques made contributions to the NLP community with the advantage of combining deep learning. These works [15-17] pushed the frontier of many typical NLP tasks and even achieved performance that better than human level in some specific pattern recognition tasks such as name entity recognition and text classification. However, it is admitted that large language model and others specified task models still have gap behind human performance in a lot of more complex fields that require understanding and creativity. Natural language generation is a research topic that highly requires model’s language structure understanding, the result of a generating model should not only satisfy language fluency but also semantics continuity. For example, text summarization is kind of text generation tasks in which a model is required to generate a certain length of summarization based on the given paragraph in the inference mode. Another example is human machine dialog system or chatbot system in which a dialog model is built to recognize and understand natural language level query and output a most relevant answer. In a more generalize perspective, a dialog model can also play a role like an intelligent assistant with the functionality that not only answering questions but also providing suggestions.

1.3Human factors in NLP

As aforementioned, even though a lot of state-of-the-art results achieved significant performance in conventional NLP tasks, there is still a large room for improvement in other more challenge text generation tasks. To make an NLP system more intelligent, [18] first proposed a concept that treating every NLP problem as a text-to-text task. Under this concept, the model understands which tasks should be performed thanks to the task-specific prefix added to the original input sentence (e.g., “translate English to German:”, “summarize:”). This is an evolutional idea which encouraged to bridge the gap of language understanding between human and a machine learning model. Another work [19] introduced few-shot setting to language model in which the language model is equip with the capability to handle unseen tasks without the need of task-specific fine-tuning. This is to model a core idea that humans generally perform a new language task from only a few examples or from simple instructions.

However, the works mentioned above are still stay in the idea that modeling human behavior in performing language tasks. In that case, no matter how large volume of the parameters the models have, unintended behaviors like not following the user instructions are always encountered. We normally call this kind of problem as model misaligned. To solve the misalignment problem in language model objective , some state-of-the-arts [20, 21] are proposed to engage human feedback in a ML model. To be specific, this idea is applying reinforcement learning (RL) to train or finetune a language model with human judgement as reward (more details are included in the later part of this paper). Particularly, this work [22] directly leads to extremely promising and evolutional project in the whole artificial intelligent but not only NLP community, that is ChatGPT, a chatbot system, from openAI. These models that engaged human judgement and preference in the training loop, indeed is taking advantage of human domain knowledge as part of the model prior experience. They significantly improved the alignment between model output and human output to a certain input.

1.4 Contents organization

In the following section 2, the details and techniques pipeline about the most state-of-the-art HITL method in NLP would be illustrated. In section 3, novel research idea and direction would be proposed based on the current works that illustrated in this paper. In section 4, conclusion would be given. The purpose of this paper is to review the current frontier of HITL for NLP community, meanwhile attempting to explore novel research direction in this area.

2. Human-in-the-loop method

There are many methods and definitions when talking about the phrase “Huma-in-the-loop”. They can be divided into three categories based on the perspective they are referring to according to this work [3], which are: 1) data processing method based on human-in-the-loop; 2) model training and reasoning based on human-in-the-loop; 3) and system construction and application based on human-in-the-loop. Under this context, we focus on the language model training and reasoning with human-in-the-loop.

In most current state-of-the-arts in NLP community, human-in-the-loop training can be summarized to applying reinforcement learning to finetune a policy (which is a pre-trained language model) with a human preference supervised training model as the reward function. This idea is specifically inspired by the work [23] in which a robot locomotion RL tasks can be optimized with an unobservable reward function that defined by non-expert users' comparison on short video clips of the agent's behavior. A high level overview can be seen on Figure 2 [23].

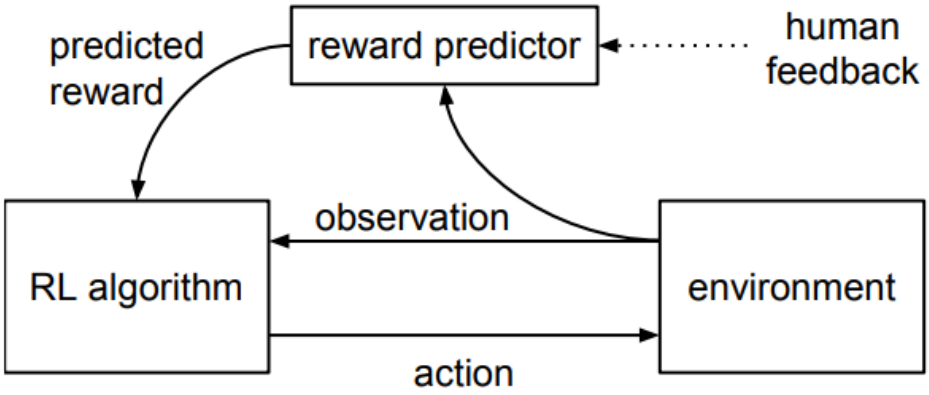


Figure 2. The reward predictor is trained asynchronously from comparisons of human feedback, and the agent maximizes predicted reward.

In general, consider an agent's actions that depend on the environment is a sequence of steps, at each time step, the agent observes a state $o_t \in \mathcal{O}$ from the environment and responds an action $a_t \in \mathcal{A}$ to the environment. In traditional RL problem, the environment will also provide a reward $r_t \in \mathbb{R}$, so the agent's goal is to maximize the discounted sum of rewards. However, in this scenario, a human supervisor is assumed to provide preferences between trajectory segments, which is a sequence of pairs of observed states and actions, $\sigma = \big((o_0, a_0), (o_1, a_1), \dots, (o_{k-1}, a_{k-1})\big) \in (\mathcal{O} \times \mathcal{A})^k$. Let $\sigma^1 \succ \sigma^2$ if trajectory segment σ^1 is preferred by human to σ^2. In this case, the agent's goal is to produce human preferred trajectories, while request as few queries as possible from human. In quantitative, we can say that the preferences $\succ$ are generated by a reward function $r: \mathcal{O} \times \mathcal{A} \to \mathbb{R}$ if,

$$\Big((o_0^1, a_0^1), (o_1^1, a_1^1), \dots, (o_{k-1}^1, a_{k-1}^1)\Big) \succ \Big((o_0^2, a_0^2), (o_1^2, a_1^2), \dots, (o_{k-1}^2, a_{k-1}^2)\Big)$$

wherever

$$r(o_0^1, a_0^1) + \cdots + r(o_{k-1}^1, a_{k-1}^1) > r(o_0^2, a_0^2) + \cdots + r(o_{k-1}^2, a_{k-1}^2)$$

If the human's preferences are generated by the reward function, the agent will have a higher reward from r. Thus, if the reward function is known, the agent can be evaluated quantitatively. Using RL to optimize r, the agent would be able to achieve the ideally high reward. However, in some cases, we are able to use a quantitative reward function to evaluate the behavior. If so, qualitative method can still be used to evaluate how well the agent satisfy the human preferences.

Further, the RL objective is to learn a good decision-making policy $\pi: \mathcal{O} \to \mathcal{A}$ with the reward function $\hat{r}: \mathcal{O} \times \mathcal{A} \to \mathbb{R}$, which maximizes rewards over time. These functions can be

parameterized by deep neural networks. The networks can be trained following the below procedures:

1. Sampling the initial policy π to generate a set of trajectories $\tau^1, \dots, \tau^i$, then update the parameters of π by some RL algorithms so that sum of the predicted rewards $r_t = \hat{r}(o_t, a_t)$ can be maximized.
2. Randomly selecting a pair of segments (σ^1, σ^2) from trajectories $\tau^1, \dots, \tau^i$ that generated from step 1 for human to compare.
3. Optimizing the parameter in r_t through supervised learning settings to fit the human preference pattern.

The above training process is run interactively. The first process generates trajectories, which are sent to the second process for comparison by a human overseer. The human comparisons then go to the third process, which uses the information to update parameters for r_t. Finally, the updated parameters are sent back to the first process to continue generating trajectories. This is the formalize pipeline for the deep reinforcement learning with human feedback based on the framework in [23]. The following part will illustrate the details about how it could be applied to settings of training language model.

2.1 Model

Suppose that an autoregressive language model ρ with a vocabulary Σ was defined by probability distribution over sequence of tokens Σ^n such that,

$$\rho(x_0 \cdots x_{n-1}) = \prod_{0 \le k < n} \rho(x_k | x_0 \cdots x_{n-1})$$

This model can be applied to a task with input space $X = \Sigma^{\le m}$, then the data distribution $\mathcal{D}$ is over X, and the output space $Y = \Sigma^n$. For example, if $x \in X$ was an article that has 1000 words, then $y \in Y$ can be a summary with 100 words. Using ρ to generate the output summary can be regarded as a sampling obey a probabilistic policy via $\rho(y|x) = \rho(xy)/\rho(x)$, in which every next token can be generated by the prior tokens.

Consider initializing a policy with a pre-trained language model such that $\pi = \rho$, then we can train π via RL with a reward function $r: X \times Y \to \mathbb{R}$ to satisfy the given task. In that case, the goal of the optimization is to maximize the expected reward,

$$\mathbb{E}_\pi[r] = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi(\cdot|x)}[r(x, y)]$$

However, rather than using a given reward, we need to estimate a reward function $\hat{r}$ which reflects the human preference in the certain task. Therefore, we have to collect the human preference on this task, and then use these preferences as the dataset to train a supervised reward model.

Following the scheme in [23], to collect the human preference data, human labelers should be asked to select a best value y_i response to a given x among two or more than two options such that $i \ge 2$. Let us continue to take the article summary task as an example. Consider for each given article, there are four options of summary (y_0, y_1, y_2, y_3). Let $b \in 0,1,2,3$ be the label of the option that selected by human. Then we can construct a dataset S that each data point as a tuple

$(x, y_0, y_1, y_2, y_3, b)$ in which b is the label for supervised learning (actually there are still other ways to construct dataset, as long as using the label information). Then the reward model $r: X \times Y \to \mathbb{R}$ can be fit using the below loss,

$$loss(r) = \mathbb{E}_{(x,\{y_i\}_i,b)\sim S}\left[log\frac{e^{r(x,y_b)}}{\sum_i e^{r(x,y_i)}}\right]$$

Given that the reword model needs to natural understand, following the implementation in [24], it can be initialized as a linear function of the final embedding layer of the language model ρ. Ordinally, the output of the model would be normalized to mean 0 and variance 1 for $x\sim\mathcal{D}, y\sim\rho(\cdot\,|x)$.

With the reward model and summary generation policy (which is also a model) well defined by neural networks. The training process is just an asynchronous training between the optimization of r and finetuning of π with RL. However, recalling the training process 1 as aforementioned, in the context language model, every epoch's training data is sampling from the generating samples from last epoch's policy. Consider in any one of the training epochs, if an RL update results in a poor policy, the future sample batches may not provide much meaningful information as it may get farther from the distribution of the original language model. Therefore, to prevent π getting too far from ρ, a KL divergence term is added to the reward as a penalty. After adding the penalty term, the RL is performed on the modified reward such that,

$$R(x, y) = r(x, y) + \beta \log\frac{\pi(y|x)}{\rho(y|x)}$$

where the d coefficient β controls the strength of the KL penalty. Worth mentioning, in an online training setting, rather than simply iterating the original three training steps, the data collection and training procedures can be updated continually over the whole project if resource is available. By constantly updating the human preferences data through the whole training, it makes the model become more human-in-the-loop as it can continually taking more human variants into consideration. The specific illustration can be seen in Figure 3 [21].

2.2 Policy gradient method

Most, if not all, RL optimization problems are based on policy approximation and value approximation. However, under the context of training language model based on human feedback, in most scenarios the optimization problem is solved by policy approximation, which is rooted on policy gradient methods. Tons of policy gradient algorithms have been proposed following the emergence of RL, it is unrealistic to exhaust them is this paper. In most current state-of-the-arts in this topic, Proximal Policy Optimization (PPO)[25] and Asynchronous Advantage Actor-Critic (A3C) [26] are the two most popular favor. But both are based on policy gradient theorem. Some brief concepts of policy gradient theorem would be given as follows.

Recall that RL problem's goal is to get a maximize expectation reward. So let us begin with the optimization problem. Consider the maximization problem (in a finite trajectory space),

$$\max_{\vartheta} J(\vartheta) = \max_{\vartheta} \sum_{\tau} P(\tau;\vartheta) R(\tau)$$

where $J(\vartheta)$ is the objective function of the RL problem; $R(\tau)$ is the reward function corresponding to different trajectories, in our setting, is the reward model based on human preferences; $P(\tau;\vartheta)$ is the trajectory probability corresponding to a certain trajectory (in our case is a function over different token combination) which can be further defined by,

$$P(\tau;\vartheta) = \prod_{t=0}^{T} P(o_{t+1}|o_t, a_t) \cdot \pi_{\vartheta}(a_t|o_t)$$

where $\pi_{\vartheta}(o_t|s_t)$ is language model (or generation policy); $P(o_{t+1}|o_t, a_t)$ is the transition probability distribution that model on the environment. In theory, the $P(o_{t+1}|o_t, a_t)$ is a troublesome because it is implicit. In practical problem, however, this term can be skip since in most policy gradient algorithms, the gradients of $J(\vartheta)$ is based on estimation methods in which the derivative of $P(o_{t+1}|o_t, a_t)$ can be removed (it is not depended on ϑ) through logarithm trick such that,

$$\nabla_{\vartheta} \log P(\tau;\vartheta) = \nabla_{\vartheta} \sum_{t=0}^{T} \log \pi_{\vartheta}(a_t|o_t)$$

Through different estimation methods, it is easy to calculate $\nabla_{\vartheta} J(\vartheta)$, and the thus the update can be performed like this,

$$\theta = \theta + \alpha \nabla_{\vartheta} J(\vartheta)$$

where α is the learning rate. In some advanced policy gradient algorithms (e.g., PPO), the α is absorbed into the estimation of $\nabla_{\vartheta} J(\vartheta)$.

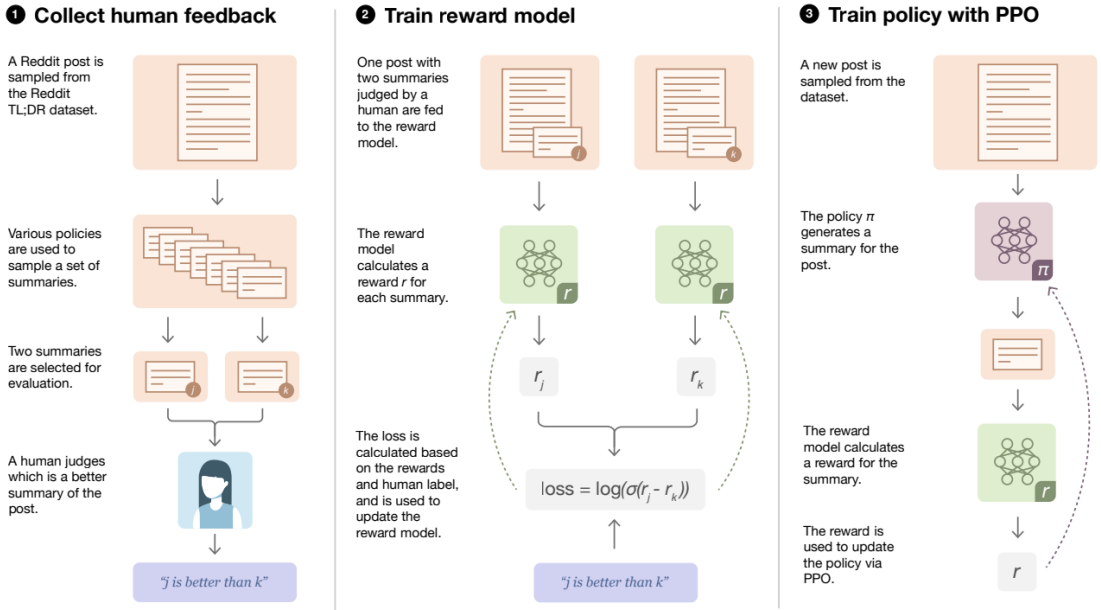
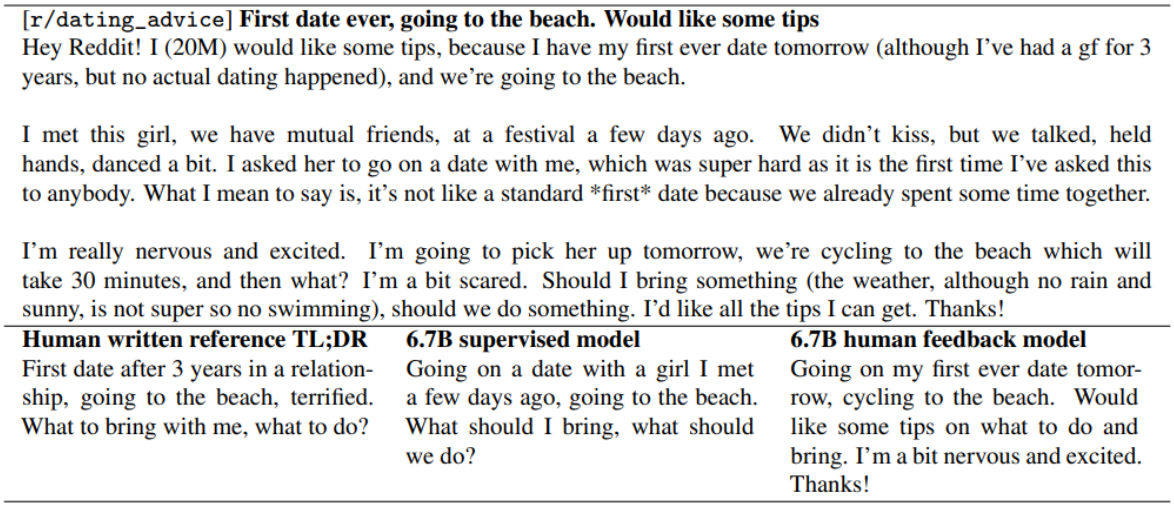


Figure 3. Diagram of our human feedback, reward model training, and policy training procedure.

3. Discussion and future works exploration

The current state-of-the-art with human-the-loop method are really impressive for the whole NLP community, especially the most representative work ChatGPT in which human feedbacks are took the most advantage to improve the intelligent level. Admittedly, these works significantly pushed forward the frontier NLP research, successfully aligning the language model capability with human intelligence in pure text tasks (an impressive result can be seen in Figure 4 [21]). However, there are still obvious limitations in these models. Firstly, even though they engaged human feedbacks in the training loop, the ML model training framework does not fundamentally change. The system is still basically following the traditional procedure: 1) data collecting; 2) model training; 3) model deploying. Some works did claim that they have some online training settings that parallel the human data collection with the finetuning process, but the change is not evolutional enough. In addition, their works' settings required human labelers to strictly follow a series of instructions, which may generate unknow impact to some human preferences . Secondly, the above discussed human-in-the-loop pipeline are most designed for single modality model. There is a large room for the application of human-in-the-loop in multimodality tasks that cut across image and text data.



[r/dating_advice] **First date ever, going to the beach. Would like some tips**
Hey Reddit! I (20M) would like some tips, because I have my first ever date tomorrow (although I've had a gf for 3 years, but no actual dating happened), and we're going to the beach.

I met this girl, we have mutual friends, at a festival a few days ago. We didn't kiss, but we talked, held hands, danced a bit. I asked her to go on a date with me, which was super hard as it is the first time I've asked this to anybody. What I mean to say is, it's not like a standard *first* date because we already spent some time together.

I'm really nervous and excited. I'm going to pick her up tomorrow, we're cycling to the beach which will take 30 minutes, and then what? I'm a bit scared. Should I bring something (the weather, although no rain and sunny, is not super so no swimming), should we do something. I'd like all the tips I can get. Thanks!

Human written reference TL;DR	**6.7B supervised model**	**6.7B human feedback model**
First date after 3 years in a relationship, going to the beach, terrified. What to bring with me, what to do?	Going on a date with a girl I met a few days ago, going to the beach. What should I bring, what should we do?	Going on my first ever date tomorrow, cycling to the beach. Would like some tips on what to do and bring. I'm a bit nervous and excited. Thanks!

Figure 4. Comparison of different summaries to a story. The left one is a summary written by human, the middle one is generated by traditional supervised learning model, the right is generated by human feedback model.

3.1 Future for HILT in multimodality task

While the purpose of this paper is to discuss and review the human-in-the-loop technique under the context of NLP. The range of research topic that the NLP community concern is not limited to the tasks that only contain text data. To be more specific, multimodality tasks that involve text and image input is a minority area that also in NLP community. For example, image captioning is a task that require a captioning model to generate text descriptions for a given input image. In the current research community, most human feedback training task are fall in article summarization, story continuation. But there is not too much works apply human-in-the-loop training in image

captioning yet due to the difficulty in aligning the two data modality. We believe that applying the human feedback-based training pipeline on this task in which using human preferences on image descriptions to train an image captioning would be an interesting work for future research.

3.2 Future for HILT in attack and defense of NLP task

While the great successes of large language model achieved in recent years motivate the research community continually push the-state-of-the-arts, the number of researchers that concern the security of large language is still small when comparing to which do research on developing new algorithms. This makes language model security still a nascent topic in NLP research. Though some prior works concern about this topic in which novel attacks and defense are proposed, the spotlight on this topic is far away from enough. The above human-in-the-loop training pipeline can be used to train adversarial attack samples for model robustness evaluation purpose. Further, it is also promising to use this technique to construct a large benchmark dataset for ML security under the context of NLP.

4. Conclusion

In this paper, we did a detailed review on the current research status of human-in-the-loop technique and its application on text generation tasks. In the introduction part, we reviewed and illustrated some basis conceptions on NLP and HITL. The core idea is to discuss the importance and promise to integrate human domain knowledge into language model. In the method section, we detailly explained main theory and principle of a popular human-in-the-loop training pipeline, which has been proved successful and used for some state-of-the-arts. Finally, the paper is end with a discussion about the limitations of current works and future research direction. We hope this paper will help readers get a little bit more familiar with human-in-the-loop training method for language model.

Reference

[1] S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications," *Computer Science Review,* vol. 40, p. 100379, 2021.

[2] A. Brutzkus and A. Globerson, "Why do larger models generalize better? A theoretical perspective via the XOR problem," in *International Conference on Machine Learning*, 2019: PMLR, pp. 822-830.

[3] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He, "A survey of human-in-the-loop for machine learning," *Future Generation Computer Systems,* 2022.

[4] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu, "Layoutgan: Synthesizing graphic layouts with vector-wireframe adversarial networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 43, no. 7, pp. 2388-2399, 2020.

[5] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient gan training," *Advances in Neural Information Processing Systems,* vol. 33, pp. 7559-7570, 2020.

[6] H. T. Shen *et al.*, "Heterogeneous data fusion for predicting mild cognitive impairment conversion," *Information Fusion,* vol. 66, pp. 54-63, 2021.

[7] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur),* vol. 53, no. 3, pp. 1-34, 2020.

[8] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Science China Technological Sciences,* vol. 63, no. 10, pp. 1872-1897, 2020.

[9] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, "A survey: Deep learning for hyperspectral image classification with few labeled samples," *Neurocomputing,* vol. 448, pp. 179-204, 2021.

[10] X. Zhang, S. Wang, J. Liu, and C. Tao, "Towards improving diagnosis of skin diseases by combining deep neural network and human knowledge," *BMC medical informatics and decision making,* vol. 18, no. 2, pp. 69-76, 2018.

[11] R. Zhang, F. Torabi, L. Guan, D. H. Ballard, and P. Stone, "Leveraging human guidance for deep reinforcement learning tasks," *arXiv preprint arXiv:1909.09906,* 2019.

[12] A. Holzinger *et al.*, "Interactive machine learning: experimental evidence for the human in the algorithmic loop: A case study on Ant Colony Optimization," *Applied Intelligence,* vol. 49, pp. 2401-2414, 2019.

[13] Y.-t. Zhuang, F. Wu, C. Chen, and Y.-h. Pan, "Challenges and opportunities: from big data to knowledge in AI 2.0," *Frontiers of Information Technology & Electronic Engineering,* vol. 18, pp. 3-14, 2017.

[14] D. Jurafsky, *Speech & language processing*. Pearson Education India, 2000.

[15] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, 2017.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805,* 2018.

[17] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[18] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research,* vol. 21, no. 1, pp. 5485-5551, 2020.

[19] T. Brown *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems,* vol. 33, pp. 1877-1901, 2020.

[20] D. M. Ziegler *et al.*, "Fine-tuning language models from human preferences," *arXiv preprint arXiv:1909.08593,* 2019.

[21] N. Stiennon *et al.*, "Learning to summarize with human feedback," *Advances in Neural Information Processing Systems,* vol. 33, pp. 3008-3021, 2020.

[22] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," *arXiv preprint arXiv:2203.02155,* 2022.

[23] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems,* vol. 30, 2017.

[24] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training."

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347,* 2017.

[26] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016: PMLR, pp. 1928-1937.