

IT2ACL: Learning Easy-to-Hard Instructions via 2-phase Automated Curriculum Learning for Large Language Models



Yufei Huang, Deyi Xiong*

College of Intelligence and Computing, Tianjin University, Tianjin, China
 {yuki_731, dyxiong}@tju.edu.cn

Continual Learning: Language models' generalization and continual learning capabilities can be enhanced by integrating Instruction Tuning with Curriculum Learning.

MOTIVATION

Curriculum Learning: The fundamental idea is to train models in an order from simple to complex instead of having the model handle all the training samples right from the start. This approach allows the model to first grasp the basics and then progressively transition to more complex tasks.

Limitations of Existing Work:

- To adhere to the curriculum learning method, it is crucial to order tasks accordingly. But Measuring task difficulty is challenging, especially in the domain of multi-task natural language generation. Prior rule-based strategies relied largely on experience and intuition. As model's capabilities enhance, longer sequences or rare tokens aren't always considered "difficult". And The complexity of the tasks and instructions that appear simple to humans might not necessarily be for models.
- Establishing a fixed order might not be optimal and manually defining the difficulty of a task is a complex and time-consuming process.

IT2ACL: we propose IT2ACL, a Instruction Tuning framework guided by 2-phase Automated Curriculum Learning, which learns easy-to-hard instructions for language models.

More specifically,

- In the first phase, the model learns different tasks in an order from simple to complex. In the second phase, once a particular task is determined, the model progresses from easier to more challenging instructions within that task.
- We introduce a progress signal to guide the order of learning for both phases. The instruction prediction gain focuses on the decrease in loss when the model trains under different instructions within the same task.
- After several training sessions, the average of all instruction signals within a task becomes the task's overall signal. Our training scheduler uses the adversarial multi-armed bandits (MAB) strategy, where the progress signal is scaled and used as the reward in MAB, enabling dynamic selection of the optimal training path to maximize gains.

METHODOLOGY

Curriculum Definition

- Set A: input data sequence
- Set B: target sequence
- Sequence T: instruction formatting
- Instances $X = (A \times T \times B)^N$
- Distribution D: Task Clusters
- Θ : model parameters
- Loss for the model on the Kth task:

$$L_k(\theta) := \mathbb{E}_{x \sim D_k} L(x, \theta) \quad (1)$$

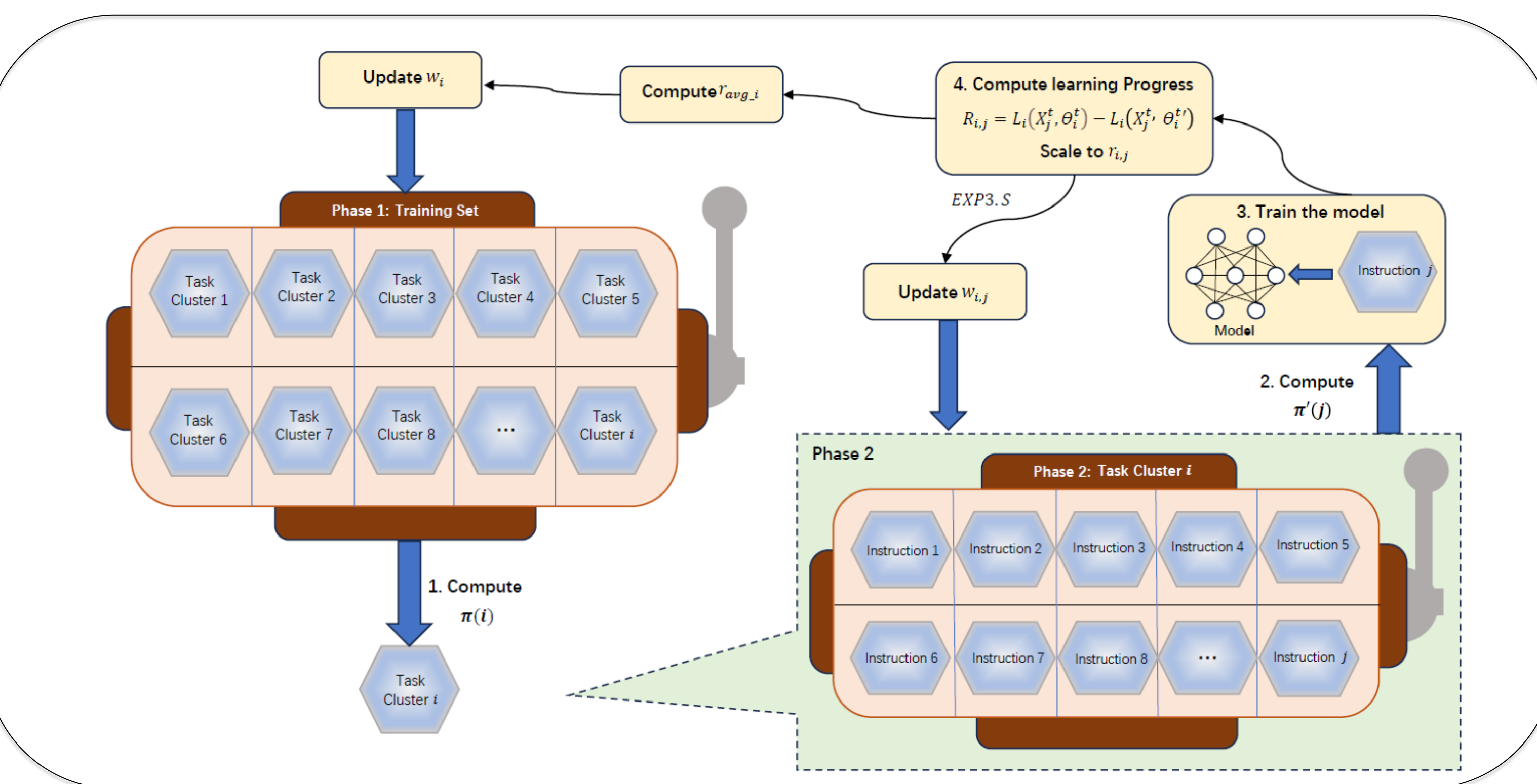
- Loss in multi-task scenario of instruction tuning:

$$L_{MT} := \frac{1}{N} \sum_{k=1}^N L_k \quad (2)$$

Instruction Prediction Gain

$$R_{INS} = L_i(\mathcal{X}_j^t, \theta_i^t) - L_i(\mathcal{X}_j^t, \theta_i^{t'}) \quad (3)$$

$$r_t = \frac{2(R_{INS} - R_{t_{min}})}{R_{t_{max}} - R_{t_{min}}} - 1 \quad (4)$$



Average Reward

$$r_{avg_k} = \frac{\sum_j R_{k,j}}{\sum_j C_{k,j}} \quad (8)$$

Adversarial Multi-Armed Bandits

$$\pi_t^{EXP3.P}(i) := (1 - \epsilon) \pi_t^{EXP3}(i) + \frac{\epsilon}{N} \quad (7)$$

$$w_{t,i}^s := \log \left[(1 - \alpha_t) \exp \left\{ w_{t-1,i}^s + \eta \tilde{r}_{t-1,i}^s \right\} + \frac{\alpha_t}{N-1} \sum_{j \neq i} \exp \left\{ w_{t-1,j}^s + \eta \tilde{r}_{t-1,j}^s \right\} \right] \quad w_{1,i}^s := 0 \quad \alpha_t := t^{-1} \quad \tilde{r}_{s,i}^\beta := \frac{r_s \mathbb{1}_{a_s=i} + \beta}{\pi_s(i)}$$

Automated Curriculum Learning For Instruction Tuning

Algorithm 1 Two-phase Automated Curriculum Learning

Initially: $w_i = 0$ for $i \in D$, $w_{i,j} = 0$ for $j \in T$, historical rewards $R_{i,j} = 0$ and counts $C_{i,j} = 0$

for $t = 1 \dots T$ **do**
 $\pi(k) := (1 - \epsilon) \sum_i \frac{e^{w_{k,i}}}{e^{w_{k,i}} + \frac{\epsilon}{N}} + \frac{\epsilon}{N}$
 Select task index k from $\pi(k)$.
 $\pi'(j) := (1 - \epsilon) \sum_j \frac{e^{w_{k,j}}}{e^{w_{k,j}} + \frac{\epsilon}{M}} + \frac{\epsilon}{M}$
 Select instruction index j for task k from $\pi'(j)$.

Train network p_θ using instruction j of task k .
 Compute learning progress R . (Sections 3.4.1)
 Scale R to reward r .
 Update $w_{k,j}$ using Exp3.S (7).
 Compute $R_{k,j}$ and $C_{k,j}$.
 Compute weighted average reward using Eq.(8)
 Update w_k using Exp3.S (7).
end for

EXPERIMENTAL RESULTS

Full-shot Results Compared to Baselines

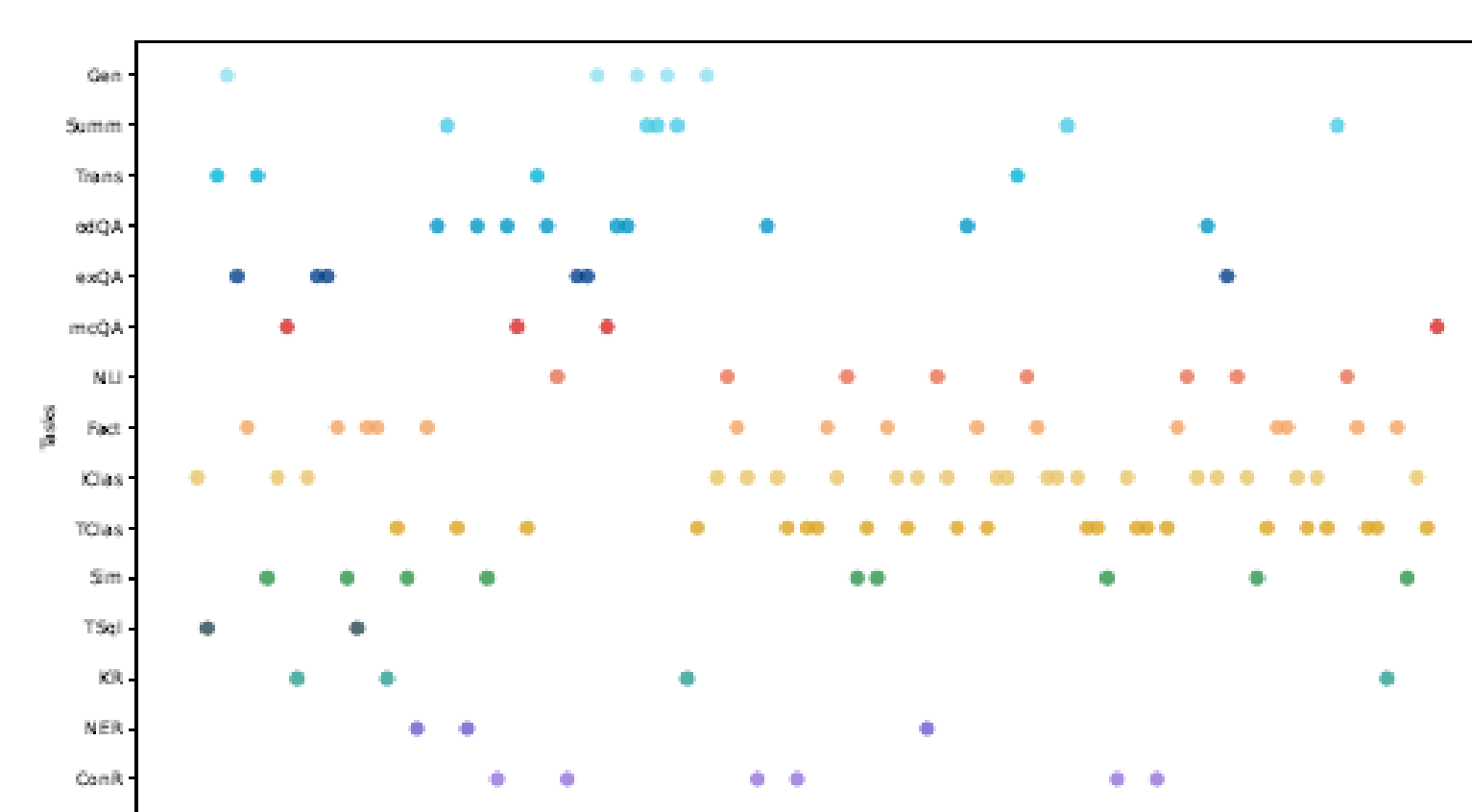
Method	Model	ConR	NER	KR	TSql	Sent	Sim	TClas	IClas	Fact	NLI	mcQA	exQA	odQA	Trans	Summ	Gen
Single Task	BART	31.5	34.6	36.1	27.7	30.9	43.8	40.2	39.6	38.1	33.4	31.8	41.2	25.3	31.5	42.7	41.9
	mT5	31.7	41.8	42.9	35.5	44.6	48.9	50.6	45.2	49.3	45.6	35.5	39.5	27.9	41.2	35.8	32.7
IT(None)	BART	25.7	26.1	21.1	23.7	18.5	41.2	34.5	37.8	30.1	35.4	16.7	40.0	23.3	16.6	39.5	38.6
	mT5	26.2	28.8	37.6	29.2	40.1	45.4	46.5	42.2	47.2	42.0	33.4	34.8	26.3	24.1	32.9	30.8
Rule-based CL(Pre-defined Order)	BART	25.2	29.8	30.3	23.8	24.9	40.7	33.2	37.3	31.5	34.2	20.5	39.8	24.4	16.9	39.2	40.7
	mT5	26.8	30.5	37.9	28.4	40.7	48.5	46.9	43.6	48.8	43.7	34.2	35.9	27.5	26.3	33.9	31.9
IT1ACL	BART	25.9	30.1	32.8	25.2	28.8	42.2	38.1	37.5	35.8	34.9	26.4	41.4	24.8	16.7	42.6	41.2
	mT5	30.2	36.2	39.3	28.8	45.5	48.7	49.2	45.7	49.2	45.5	35.6	36.1	28.3	25.8	34.3	32.5
IT2ACL(ours)	BART	26.6	31.5	32.4	25.7	30.7	42.5	39.5	37.6	36.1	33.4	29.3	42.9	25.1	16.8	43.9	42.3
	mT5	30.4	37.9	39.8	29.1	45.2	48.5	49.7	46.8	49.6	46.3	34.1	36.4	29.1	25.6	34.7	33.1

Table 1: Overall experimental results of our proposed framework IT2ACL and the compared baselines. The columns denote different tasks: **ConR** stands for Conference Resolution, **NER** for Named Entity Recognition, **KR** for Keywords Recognition, **TSql** for SQL Task, **Sent** for Sentiment Analysis, **Sim** for Similarity, **TClas** for Text Classification, **IClas** for Intent Classification, **Fact** for Fact Checking, **NLI** for Natural Language Inference, **mcQA** for Multiple Choice Question Answering, **exQA** for Extractive Question Answering, **odQA** for Open-Domain Question Answering, **Trans** for Translation, **Summ** for Summarization, and **Gen** for Text Generation.

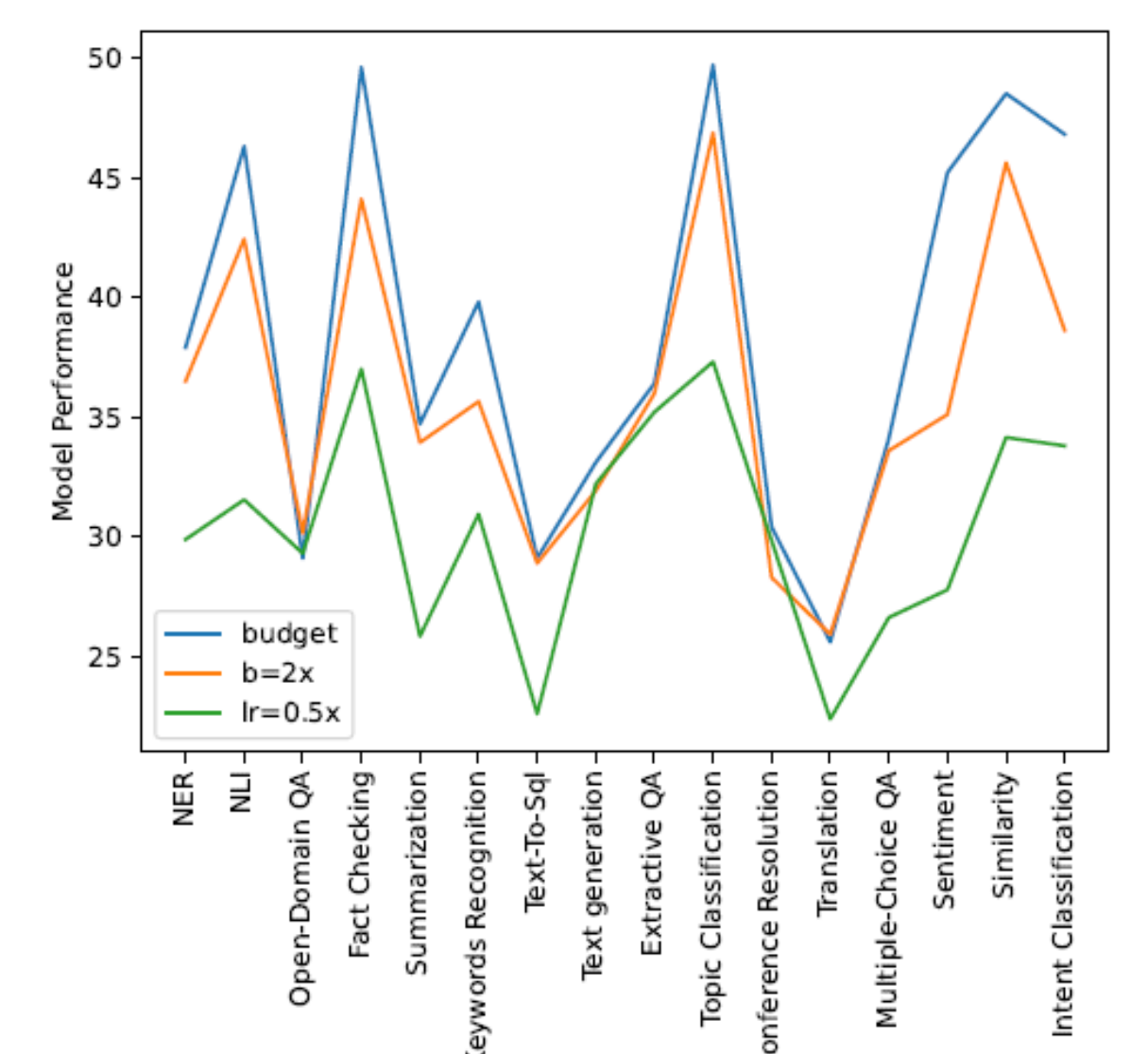
Generalization Results Compared to Baselines

Method	Model	NLI	mcQA	Summ
IT(None)	BART	24.8	18.9	32.9
	mT5	29.5	17.9	25.9
Rule-based CL(Pre-defined Order)	BART	22.6	20.8	31.2
	mT5	28.7	19.8	24.6
IT1ACL	BART	25.4	21.3	35.7
	mT5	31.0	22.3	27.1
IT2ACL	BART	26.9	22.8	36.9
	mT5	32.2	23.5	28.3

Sampling frequency of different tasks during training process



Effect of hyperparameters settings



Change of generalization ability over the course of training

