Lite Manual: Chain-of-Thought Enhanced Step-by-Step Summarization with Knowledge Distillation

Group 6

Abstract

Large Language Models (LLMs) with in-context learning (ICL) abilities can successfully generate accurate summarization results using innovative prompting techniques, but they demand significant computational resources. Smaller language models require fewer computational resources, but they rely on large labelled datasets for fine-tuning to achieve desired accuracy. The challenge lies in addressing the constraints of limited labeled data and computational resources simultaneously. To tackle this problem, we present a novel method that leverages the ICL capabilities of LLMs with inventive prompting techniques for high-quality instructional summaries. Our approach uses K-means clustering for prompt selection balance, a CoT table to boost LLM reasoning, and effectively transfers knowledge from a teacher to a smaller student model. This method optimizes resource use and improves performance, while addressing challenges arising from limited resources and data in NLP.

1 Introduction

The success of Large Language Models (LLMs) like GPT-4 has led to significant advancement in performance across various Natural Language Processing (NLP) tasks such as long text summarization (OpenAI, 2023). While LLMs are incredibly powerful, they may not be suitable for specific domains or tasks. Rather, it is often necessary to fine-tune a pre-trained language model to cater to a specific domain of knowledge or task requirement. A smaller language model is more feasible to be fine-tuned, requiring less computational resource and time to train and draw inference. However, labeled data are often required but limited.

Addressing the challenge of limited labeled data and computational resources, we propose a novel approach on the WikiHow dataset (Koupaee and Wang, 2018) for generating step-by-step instructional summaries using in-context learning (ICL) abilities of LLMs through various innovative prompting engineering techniques. We explore various prompting strategies, highlighting the importance of balancing diversity and similarity in prompt selection. Furthermore, we introduce a chain-of-thought (CoT) table to enhance the LLM's reasoning ability. Finally, we conduct knowledge distillation from a teacher model LLM to a smaller pre-trained student model, enabling the student model to generate summaries for a manual dataset. Although the student model performs well as a lite manual generator, it still has room for improvement compared to the teacher model. The CoT dataset consistently outperforms other ICL datasets.

1.1 Contributions

These are the main contributions of our paper:

- K-means clustering is employed to balance the diversity of prompt demo examples, enabling more effective prompt selection by considering both similarity and diversity within the dataset.
- An innovative Chain-of-Thought (CoT) table is introduced to highlight the causal relationships, actions, and objectives of each step in an instruction, thereby enhancing the reasoning ability of the language model when generating summaries. It is important to note that prior work has rarely utilized the CoT approach in downstream summarization tasks, possibly due to the challenges associated with achieving universal zero-shot CoT generation and manual labeling.
- The impact of the number of steps in instructions on the LLM's reasoning ability is investigated, providing insights into the model's performance in relation to instruction complexity.

- Successful implementation of manual summarization ability transfer from a teacher model to a student model is demonstrated, showcasing the effectiveness of the approach in efficiently distilling knowledge between models.

2 Related Work

This section reviews research that lays the foundation of our work:

In-context learning: ICL (Dong et al., 2022) is a few shot learning technique that allows LLMs to learn from a few examples in the context without gradient update. LLMs leverage ICL abilities when prompted a few examples as the input (Brown et al., 2020). Studies have shown that the strength of ICL varies widely depending on the in-context demonstrations (Liu et al., 2021).

Chain-of-thought: In the article Wei et al. (2022), the concept of chains of thought is introduced as a form of discrete cue learning for contextual learning in LLMs. Building upon this, Wang et al. (2022b) enhances the performance of the approach through majority voting on answers. Concurrently, Zelikman et al. (2022) proposes a boosting method that enables even small and medium-sized models to be trained to have thought chain capabilities. Kojima et al. (2022) presents a method that simplifies complex problems into easier sub-problems, leading to improved performance compared to the original zero-shot, few-shot, and CoT approaches. This idea is extended by Huang et al. (2022), which combines CoT and Self-Consistency by using majority voting to obtain answers and fine-tuning language models with solutions generated by the models themselves, thus enabling self-improvement. In Zhang et al. (2022), the gap between Zero-shot-CoT and Few-shot-CoT is bridged by using Zero-shot-CoT to build examples of Few-shot-CoT. The method involves clustering problems in the training set, sorting them by distance from the cluster center, and constructing examples for Few-shot-CoT prompts.

Knowledge distillation: Knowledge distillation (Kim and Rush, 2016) is a technique to transfer a large teacher model's knowledge down to a smaller student model using targets generated by the teacher model. The student model learns to mimic the behaviour of the teacher model to improve its performance and adapt to task-specific requirements. Fine-tuning the student model using hard target produced by the teacher model has shown to effectively improve the student model's performance (Nityasya et al., 2022).

3 Method

Our objective is to transfer the ability of manual summarization from a teacher model to a candidate student model. This process can be divided into two parts: first, we will use various prompting techniques to generate high-quality summarized data; second, we will fine-tune the pre-trained student model to effectively incorporate the knowledge provided.

3.1 ICL/COT data augmentation

Problem setup: Given a training demonstration set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, consisting of input-output sequences, where x_i represents step-by-step instructional information and y_i denotes the manual summarized information for x_i. Examples of training demonstrations can be found in the demo section of Appendix.

In this stage, it is necessary to develop a high-quality demo example retriever for the purpose of generating in-context learning data. The final prompt structure will be represented as $R\left(x_{\text{test}}, \mathcal{D}_{\text{demo}}\right)$, where x_{test} refers to the raw, unsummarized message and $\mathcal{D}_{\text{demo}}$ retrieves a subset of the training examples $\{(x_j, y_j)\}_{j=1}^{m} \subset \mathcal{D}$, where $m \ll n$. An effective retriever will imitate the in-context learning capacity of the pre-trained large language model and produce improved answers that follow the established pattern.

3.1.1 Few-shot learning random-retrieval

Initially, we introduce the most commonly used selection method, the random-retriever, which randomly selects the demo examples as prompts. This method serves as the reference group.

It is important to note the number of demo examples, m, used in this paper. Previous works, such as the CoT work by Wei et al. (2022), utilized 8 prompt examples to achieve optimal performance. However, as a result of the current language models' relatively constrained token input settings and to maintain the focus on the retriever discussion, the maximum API token limit is restricted to under 4000 tokens, as exemplified by GPT-3. We only consider m values of 1 and 2 in this study. The results of the few-shot random retrieval prompt examples can be found in Appendix Tables 9 and 10.

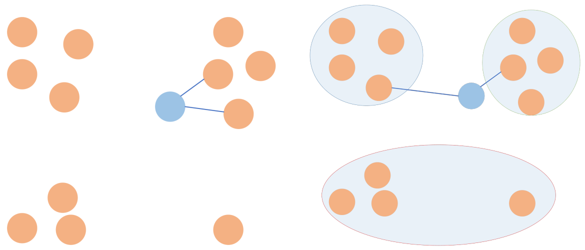

Figure 1: Demonstration of similarity-retrieval (left) and autoprompt-retrieval (right) mechanism. Blue dot denotes the test message that needs to be summarised and orange dots denote the demo examples that are used in ICL prompt construction

.

3.1.2 Auto-prompt

Similarity: Intuitively, the next logical approach is based on the similarity (Nie et al., 2022) between the question message and the demo messages. In other words, we attempt to utilize the isomorphism within the prompt construction. To achieve this, we utilize the method described in (Reimers and Gurevych, 2019) and (Rubin et al., 2021), which employs a pre-trained Sentence-BERT model for representation learning.

In detail, the Sentence-BERT model converts each text sentence into a 787-dimensional numeric vector, allowing us to calculate the Euclidean distance as a measure of sentence similarity. As a result, in the two-shot similarity-retrieval method, we select the two most similar demo examples for each prompt. The two-shot similarity-retrieval prompt examples can be found in Appendix Table 11.

Trade-off between similarity and diversity: inspired by the ideas presented in (Nie et al., 2022) and (Zhang et al., 2022), relying solely on similarity may result in poor performance in summarization, as it places too much emphasis on matching unrelated text messages or single summarization formats, reducing the generality of the in-context learning. To address this, we aim to balance diversity and similarity within the prompt examples.

To achieve this, after converting the manual text messages into vectors, as previously described, we apply the K-means algorithm to cluster the demo examples into k groups. Then, we select the k most similar prompts among each group and finally choose the top 2 as the input. The results of the two-shot autoprompt-retrieval examples can be found in Appendix Table 12.

CoT Table Zero-shot prompt question
[Original demo example text] + 'Please generate the reasoning process based on the message, refine it into a table for what action (col 1) needed to result (col 2) for what purpose (col 3) and relation to last action(parallel or causal)(col 4): '

Table 1: CoT Table Zero-shot prompt question format

Comparison: Figure 1 effectively illustrates the difference between similarity-retrieval and autoprompt-retrieval methods. In subplot (a), the retriever only selects the two nearest examples, while subplot (b) shows that the autoprompt-retrieval method selects the two nearest demo examples from two different clusters, providing a degree of diversity.

3.1.3 CoT Table

After selecting the optimal prompting demo example, we utilize the Chain-of-Thought (CoT) approach to enhance the reasoning ability of the in-context learning in language models. It is important to note that the application of CoT to downstream summarization tasks is relatively rare, as it inherently lacks the requirement for and difficulty in generating a stable chain-of-thought multi-step reasoning process and manually labeling summarization is relatively costly and prone to subjectivity.

However, due to the special multi-step instructional message structure on the Wikihow dataset, we introduce a novel CoT self-ask paradigm (Zhou et al., 2022) to perform zero-shot CoT generation (Kojima et al., 2022). In detail, for each demo example, we will combine it with an articulated prompt question, as shown in Table 1, to generate the so-called "chain-of-thought table", a new text reasoning structure that demonstrates the step-by-step language body, purpose, and relationship with the previous step sentence. A specific CoT table example can be viewed in Table 3, and the results of the two-shot autoprompt-retrieval + CoT table prompt example can be found in Appendix Table 13.

3.2 Fine-tuning

To tailor the student model for the summarization dataset, fine-tuning is performed on the pre-trained student model, leveraging the knowledge inherited from the pre-training dataset and adapting it to a

Instructional Text from WikiHow
How to Ride a Dragon in Skyrim: 1. Find a dragon: Roam around the open world of Skyrim. Keep an eye out on the sky for any signs of a dragon flying. Once you find one, go towards the direction where you see the dragon. 2. Use the Bend Will shout: Once you're near the dragon, open your in-game menu and select "Magic." Choose "Shouts" from the menu panel on the right-hand side of the game screen and select "Bend Will." 3. Mount the dragon: Approach the dragon and press the "Ride" button prompted on the game screen to mount the beast. Once you're on its back, the dragon will automatically take flight. 4. Control the dragon while flying: You can steer the beast by pressing the direction keys on your controller. You can go either left or right, upward or downward. 5. Have the dragon attack targets: While you're riding a dragon, you can order it to attack targets using its fire breath. Once you find a target, press and hold the "Lock" button on your game controller to fix your camera focus on the mark. 6. Dismount: Once you've arrived at your location or you're ready to get off, press the "Land" button on your controller and the dragon will slowly descend to land. Once it touches the ground, your character will automatically dismount the dragon and go back to walking.

Table 2: Instructional text example from Wikihow (Web link) : **How to Ride a Dragon in Skyrim**. It contains 6 steps in total to instruct the public in detail.

Step	Action	Result	Purpose	Relation
1	Roam around the open world	Find a dragon	Locate a dragon	Parallel
2	Open in-game menu	General tab appears	Access settings	Causal
3	Approach dragon	Mount dragon	Mount dragon	Causal
4	Press direction keys	Steering	Control dragon while flying	Causal
5	Press Lock button	Fix camera focus	Target attack	Causal
6	Press Land button	Descend	Dismount dragon	Causal

Table 3: Constructed Chain-of-thought table for the given **How to Ride a Dragon in Skyrim** Wikihow instructional example in Table 2

specific task (Dodge et al., 2020). This approach offers computational and data efficiency compared to traditional training methods from scratch while achieving better performance.

3.2.1 Experiment setting

GPT-3 Curie and Davinci are chosen as the teacher model, which exhibits outstanding performance in text summarization and is used for generating the ICL dataset, aiming to create structured and coherent text to serve as the ground truth dataset for knowledge distillation to train the student model. T5 is chosen to be the student model due to its applicability to various NLP tasks. In this experiment, two T5 models are employed: "T5-base" , with 220 million parameters, and "T5-small", containing approximately 60 million parameters (Raffel et al., 2020). The choice of these two models aims to investigate the effect of parameter size on the experiment's results. Both models share a similar overall architecture based on the Transformer framework, differing in the number of encoder/decoder blocks, which reflects the parameter count. The "T5-base" model contains 12 blocks, while the "T5-small" model has only 6 blocks.

The student model is initialized with pre-trained parameters, and all model blocks except the last decider block are frozen for faster training and to prevent overfitting given limited training data. These trainable parameters are then fine-tuned using ground truth labeled data and synthetic data. In this experiment, the T5 tokenizer is used to ensure that all sentences remain within a reasonable range for training, with the maximum token length for original articles set to 700 tokens and the maximum token length for summarized highlights limited to 300 tokens.

The number of epochs for training is set to 5, based on the observation that the validation loss converges within this range. A batch size of 16 is used for training both models due to hardware limitations, specifically GPU memory capacity. For the baseline model trained only by the ground truth dataset, data is split into 33% for training, 33.5% for validation, and 33.5% for testing. The training dataset is intentionally limited, assuming a scenario with limited training data resources where the ICL prompting method can be used for data augmentation. Models distilled from a teacher model have data generated from the teacher model added as extra training data. After modification, the new ratio becomes 7.7% for testing and validation and 84.6% for training.

The learning rate for the fine-turning student model is selected using grid search. A list of potential learning rates within the range of 1e-3 to 1e-6 is considered, followed by training the model and monitoring the validation loss. Consequently, a constant learning rate of 5e-5 is used for fine-tuning. A sequence-to-sequence trainer is used for fine-tuning purposes, utilizing a cross-entropy loss function as the optimizer.

4 Empirical Evaluation

4.1 Datasets

The WikiHow dataset comprises various articles that serve as manuals to guide people through different types of tasks. Each article is accompanied by a corresponding summary. This dataset has a reasonable average sentence length of approximately 100.68 words, and an average compression ratio of 2.38 falls within a suitable range, providing comprehensive summaries without being overly verbose (Koupaee and Wang, 2018).

Since WikiHow articles function as manuals and present solutions through several steps, they exhibit a natural progression of ideas or a "chain of thought". One example, shown in Table 2, illustrates how to ride a Dragon in a game called Skyrim, with six interconnected steps. It starts by finding a dragon and using magic called "Shouts", and ends by dismounting the dragon using the "Land" button. Hence, the dataset's logical flow of ideas makes it suitable for text summarization tasks using CoT.

4.2 Evaluation metric

The performance consists of comparing multiple aspects of the generated and reference texts using a combined choice of evaluation metrics. ROUGE-1 and ROUGE-2 measure the unigram and bigram overlaps, respectively, while ROUGE-L considers the longest common subsequence between the output and reference. ROUGE-Lsum, an extension of ROUGE-L, combines sentence-level and summary-level information for evaluation (Lin, 2004). BLEU (Papineni et al., 2002) is a widely-used metric in machine translation, which calculates the geometric mean of modified n-gram precision, penalizing generated texts that are too short or lack variety. METEOR (Lavie and Agarwal, 2007) is another metric considering both precision and recall, incorporating exact word, stem and synonym matches. Lastly, Gunning fog index (Gunning et al., 1952) is a measure of readability calculated as ratios of word to sentence and word to word.

For concise representation, scores of ROUGE, BLEU and METEOR with range between 0 and 1 are rescaled as percentages throughout this report.

5 Results

5.1 ICL generated data evaluation

The experiment results for generating training data using different ICL prompting techniques are compared and summarized in Table 4. The summarization generated by ICL with One-shot Random prompt produces lower performance across all evaluation metrics comparing with ICL with Two-shot Random prompt. This observation aligns with previous research, which has shown that utilizing more shots provides more contextual information and typically leads to better results in LLMs.

However, the summarization generated by ICL with Two-shot Similarity prompt resulted in worse performance than ICL with Two-shot Random prompt. This could be attributed to the lack of diversity in the prompts, which in turn leads to poorer performance. If we incorporate diversity by using K-means clustering to select the demonstration examples in the prompt, the summarization generated by ICL with Two-shot Autoprompt outperforms those created with random or similarity-based prompts. Moreover, when we introduced CoT Table into the prompting process, the results were further improved. This finding suggests that incorporating intermediate reasoning steps can enhance the ability of LLMs to generate more accurate summaries, particularly when dealing with data that possesses complex internal logic.

5.2 Fine-tune data evaluation

Table 5 presents the results of the fine-tuning performance of eight models, evaluated using vari-

Prompting Engineering	Rouge-1	R-2	R-L	Bleu	Meteor	Gunning
One-shot Random	63.7	44.9	63.3	34.1	57.9	1.76
Two-shot Random	65.2	46.1	64.7	35.3	58.8	1.80
Two-shot Similarity	64.9	45.7	64.4	34.7	58.2	1.81
Two-shot Autoprompt	66.1	47.4	65.7	**36.8**	59.7	1.81
Autoprompt + CoT Table	**67.2**	**48.1**	**66.8**	36.5	**61.5**	**1.82**

Table 4: Evaluation table for the ICL generated instructional summary data. Note: there is no standard deviation due to the training cost of LLM Api.

ous evaluation metrics. Four of these models are T5-Base models, while the remaining four are T5-Small models. Each model is fine-tuned using a different dataset, with the GT dataset acting as the baseline. To ensure the accuracy of the experiment, each model's test dataset was run five times to calculate the mean and standard deviation.

For both the T5-Small and T5-Base models, those fine-tuned with the additional ICL data consistently outperformed the baseline. This demonstrates that ICL data is highly effective for data augmentation when dealing with a limited training dataset. Table 7 and 8 in the Appendix show an example how additional high quality ICL data improves student model's fine-tuned summarization output. However, when comparing the results generated from Table 4, where the evaluation metrics values for the teacher model are shown, it is evident that there is still substantial room for the student models to improve. Using the BLEU score as an example, the teacher model consistently outperforms both student models by around +15.

Next, the effect of different datasets on the fine-tuning process can be considered. Across both model types, the performance of the CoT dataset in each evaluation metric is consistently higher than that of the K-means and one-shot. However, the differences are relatively small. Using the T5-Small model's Rouge-1 score as an example, the CoT one is only +0.3 higher than the K-means one, and the K-means dataset has a +0.4 Rouge score compared to the one-shot dataset. This result is expected, as the evaluation metric scores in Table 4 already indicate that the CoT dataset performs best. Moreover, a better dataset is essential for producing a superior fine-tuned model.

Comparing the two model types, the overall performance of the T5-Small model is better than the T5-Base model. For instance, the Rouge score of the CoT fine-tuned T5-Small model is approximately +1.9 higher than the T5-Base model. In comparison to the changes mentioned in the previous paragraph regarding the different datasets, this difference is considerably more significant. The underlying reason for this could be that only the last block of each weight for both models is made trainable, and the ratio of trainable parameters to total parameters is larger for the T5-Small model than the T5-Base model. The first 11 blocks of the T5-Base model still retain their pre-trained weights, which might negatively impact the results.

Figure 2 aims to show the evaluation metric's performance for each epoch. To do this, eight models are chosen as examples to observe how the evaluation metrics improve during training. Two metrics, Rouge-L and BLEU score, are selected for better visualization. The top two subplots in Figure 2 display the general trend for the T5-Base model, while the bottom two diagrams illustrate the T5-Small model. From the diagrams, it is obvious that both evaluation metrics exhibit similar trends with each other. For instance, in epoch 4 for training the T5-Small model, both the BLEU score and Rouge-L tend to drop. The overall performance varies with epoch number for the T5-Base model appears to be more stable than that of the T5-Small model. The CoT dataset consistently performs best among the three ICL datasets, and the Kmeans dataset is generally better than the one-shot dataset. One exception can be found in the diagram for training the T5-Small model and evaluating it using the BLEU score, where the lines for the Kmeans dataset and the one-shot dataset fluctuate quite a bit, making it difficult to distinguish which one is better. The reason for this fluctuation could be related to the number of parameters in the smaller model.

6 Ablation Experiment

6.1 Ablation Settings

We conduct a comprehensive ablation study to investigate the impact of instruction complexity present in the CoT-generated instruction data on

Model	Data	Rouge-1	R-2	R-L	R-Lsum	Bleu
T5-Small	GT	43.6±0.11	22.7±0.12	31.5±0.10	39.3±0.11	9.5±0.08
	GT+ICL(one-shot)	51.9±0.08	33.5±0.12	43.2±0.07	48.6±0.08	19.4±0.06
	GT+ICL(Kmeans)	52.3±0.16	33.9±0.14	43.6±0.12	48.8±0.16	19.8±0.08
	GT+ICL(CoT)	**52.6±0.06**	**34.1±0.07**	**43.9±0.06**	**49.2±0.04**	**20.0±0.06**
T5-Base	GT	34.3±0.18	13.5±0.12	22.5±0.12	30.2±0.17	3.2±0.07
	GT+ICL(one-shot)	50.1±0.14	30.5±0.12	40.8±0.11	46.2±0.12	16.5±0.08
	GT+ICL(Kmeans)	50.5±0.17	30.8±0.16	41.1±0.13	46.7±0.16	16.4±0.11
	GT+ICL(CoT)	**50.7±0.12**	**31.0±0.11**	**41.3±0.13**	**46.7±0.12**	**17.0±0.07**

Table 5: Evaluation performance for the generated instructional summary data from the fine-tuned student model based on different parameter scales and synthetic ICL data. Note: the mean and standard deviation values are calculated from 5 test runs on different random initializations.

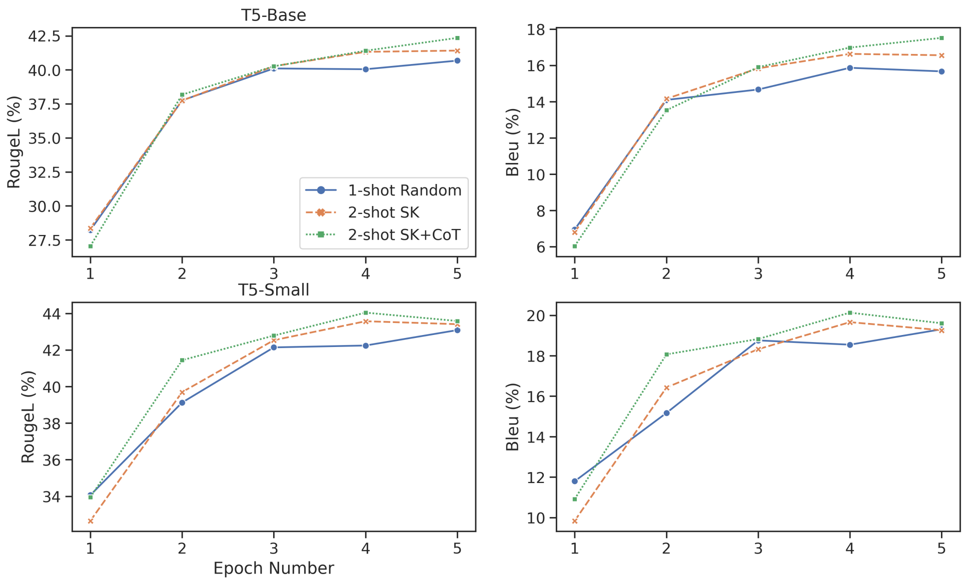

Figure 2: Evaluation performance plot for the generated instructional summary data from fine-tuned student model T5-base (top) and T5-small (bottom) under 5 epochs. Here we select the Rouge-L and Bleu as the representative metrics to highlight the improvement brought by CoT table prompt engineering.

the reasoning capabilities of large language models (LLMs). To execute this experiment, we examine the effect of training data, containing varying degrees of instruction complexity within the Wikihow problem instances synthesized by the LLM using ICL(CoT), on the fine-tuned performance exhibited by the student model.

The Wikihow problem instances have varying number of steps in the instructions, leading to differing levels of complexity and reasoning depths within the CoT instruction data. To account for these differences, we categorized the data synthesized using LLM CoT into three distinct difficulty levels according to the number of steps: 1-4 steps, 5-7 steps, and 8+ steps. We also compare these three groups with a randomly sampled group with the same number of instances. This division of data allows us to capture a broad range of difficulties while maintaining a manageable number of groups for analysis. We conduct three ablation settings to examine the fine-tuned model's performance separately for training on synthesized ICL(CoT) data at each of these three difficulty levels, and also compare them to the random setting. The fine-tune setting is the same as in section 3.2.

6.2 Experiment Results

The result of the experiment is shown in Table 6. From the score of the evaluation metrics, we find that a smaller number of steps in the LLM ICL(CoT) synthesized training data leads to better performance on the fine-tuned model. A possible explanation for this observation is that more complex instances possess higher instruction complex-

Data	Rouge-1	R-2	R-L	R-Lsum	Bleu	Flesch
ICL(CoT) Random	49.2	29.9	39.4	44.9	15.1	87.3
ICL(CoT) 1-4 steps	**52.0**	**32.6**	**42.4**	**48.0**	**16.9**	87.4
ICL(CoT) 5-7 steps	50.9	32.5	41.6	46.9	16.3	**87.6**
ICL(CoT) 8+ steps	50.1	31.5	41.0	46.3	15.8	87.2

Table 6: Model performance using training data synthesised using CoT with different number of instruction steps

ity and a longer chain-of-thought. This results into a diminished reasoning ability for the LLM, and thus poorer quality data generated by the model.

In summary, our analysis demonstrates that training the model on less complex instances with fewer steps yields better performance. This suggests that higher instruction complexity and extended CoT may negatively impact the reasoning abilities of large language models. (Wang et al., 2022a)

7 Conclusion and Discussion

In this study, we proposed an innovative and powerful prompt engineering technique, which enhances summarization reasoning ability by sorting out the action, purpose, and relation within each step in a “chain of thought table”. We also employed K-means clustering to balance the diversity and similarity among the example prompts.

Moreover, we investigated the impact of the number of steps in a message on the reasoning connection strength and found that long-step instructions could deteriorate the logic link and lead to poor performance in in-context learning with CoT.

Once high-quality training data was generated through these techniques, we proceeded to flexibly fine-tune downstream T5-base and T5-small models by adjusting the model parameters to better adapt to the specific task at hand. Our results show that the knowledge distillation process becomes more effective by combining in-context learning and prompt engineering with flexible fine-tuning. These techniques enable us to tackle the challenges posed by limited data availability, ensuring more robust and accurate outcomes in various NLP tasks.

7.1 Future work

The results of our experiments have shown that more complex instances can lead to a decrease in reasoning ability for the LLM CoT process. To address this issue, future work should consider using Graph Neural Networks (GNNs) to better capture relationships across steps and improve the CoT table generation for longer step instances. Additionally, when fine-tuning the student model, advanced techniques such as discriminative fine-tuning (Howard and Ruder, 2018) could be incorporated instead of using a constant learning rate and fine-tuning only the last layers. This could lead to improved results on the student model.

Previous work (Wang et al., 2022a) has shown that coherence is a crucial factor for the performance of CoT prompting. Further experimentation, such as an ablation study, can be carried out to investigate the impact of coherence under our specific setting. Finally, (Hinton et al., 2015) demonstrated that soft targets provide more information per training case compared to hard targets when distilling knowledge from the teacher model to the student model. In our experiments, we used hard targets, and future work could include incorporating soft targets to more effectively train the student model.

7.2 Limitations

One limitation of our experiments is the token limitation (max 2049 tokens) of the GPT-3 Curie and Davinci models, which prevented us from feeding in ICL prompts with more than two shots. While we still demonstrated the effectiveness of balancing similarity and diversity in the prompt and adding the CoT table, incorporating more shots could further improve the label quality synthesized by LLMs. Another limitation is existing automatic text summarization evaluation metrics. Although we conducted random sampling to evaluate our generated output, more thorough human evaluation experiments comparing our model summaries would provide a more comprehensive assessment of the performance of our approach.

Acknowledgement

We would like to give special thanks to Laura for her invaluable guidance and insightful feedback.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Robert Gunning et al. 1952. Technique of clear writing. *None*.

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

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. *WMT@ ACL*, 2007:228–231.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Feng Nie, Meixi Chen, Zhirui Zhang, and Xu Cheng. 2022. Improving few-shot performance of language models via nearest neighbor calibration. *arXiv preprint arXiv:2212.02216*.

Made Nindyatama Nityasya, Haryo Akbarianto Wibowo, Rendi Chevi, Radityo Eko Prasojo, and Alham Fikri Aji. 2022. Which student is best? a comprehensive knowledge distillation exam for task-specific bert models. *arXiv preprint arXiv:2201.00558*.

OpenAI. 2023. Generative pre-trained transformer 4: Language model and architecture. Technical Report TR-2023-001, OpenAI, San Francisco, CA, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2022a. Towards understanding chain-of-thought prompting: An empirical study of what matters. *arXiv preprint arXiv:2212.10001*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Appendix

Generated Instructional Summary (GT)
How to Adjust Exposure in Photoshop: 1. Open an image; the color mode of your image is RGB mode, by this my image are RGB channel, Green channel, Red channel. Example when adjust exposure by use RGB Mode. 2. xample when adapt the Exposure by Use Green mode 3. Xaplo when Adjust the output level from slide bar. Examen when use Green Mode 4. Y-xpatra when

Table 7: T5 model fine-tuned on Ground Truth data summarization output for the example **How to Adjust Exposure in Photoshop** from WikiHow, the example here is selected randomly from the Wikihow list and not included in the model fine-tuning data. The generated result is relatively obscure and verbose.

Generated Instructional Summary (CoT)
How to Adjust Exposure in Photoshop: 1. Go to Image > Adjustments > Levels. 2. Select the channel to adjust the exposure. 3. Open an image. 4. Go into Image > Adjustments > Levels, in the Level Options box, choose the channels to change the level. 5. Select input and output level from slide bar.

Table 8: T5 model fine-tuned on Ground Truth and CoT data summarization output for the example **How to Adjust Exposure in Photoshop** from WikiHow, the example here is selected randomly from the Wikihow list and not included in the model fine-tuning data. The generated result is relatively concise and clear.

Part	One-shot Random-retrieval Prompt Text
Demo1	[Message]: How to Extend a Partition in Ubuntu: 1. Open GParted: If you don't have GParted downloaded, you can get it from https://gparted.org/livecd.php. Otherwise, GParted will be located in the Dash in Ubuntu Live environments. 2. Click the partition you want to resize from the menu in the upper right corner: You can't edit partitions that are actively in use. If a partition is mounted, unmount it by clicking Eject in the file manager. 3. Right-click a partition and click Resize: When you right-click, a menu will appear next to your cursor and the resize button is usually the second option in the list. 4. Click and drag the handles on either side of the bar to expand the space: If you aren't able to make the space larger, you might need to repeat these steps to shrink a different partition first. 5. Click Resize/Move and Apply: These changes will be added to a queue that you can watch via a progress bar. This process may take a few seconds or a few minutes, depending on the number of changes that you've queued. 6. Restart your computer: You should notice the new changes once your computer restarts and boots up.
	[Summarisation]: How to Extend a Partition in Ubuntu: 1. Open GParted. 2. Click the partition you want to resize from the menu in the upper right corner. 3. Right-click a partition and click Resize. 4. Click and drag the handles on either side of the bar to expand the space. 5. Click Resize/Move and Apply.
Ask	[Message]: How to Use the Red Eye Tool in Photoshop: 1. Put the image that you want up in Photoshop. While editing, you want it as large as you can get it.; 2. In the toolbox, right click on the plaster (band-aid). Click on the icon that looks like an eye and is labeled "Red eye tool". 3. Click the eyes on your image in order to remove the red eye. You will know when you have done this successfully because you will notice the eyes are no longer flashy and the image looks much better. 4. You have now removed the redeye and you can now do what you want with the image, get it framed, put it on Facebook, etc. Be sure to downsize it when you are finished.
	[Summarisation]: ?

Table 9: One-shot in-context learning prompting example to generate the instructional summary answer of **How to Use the Red Eye Tool in Photoshop** from WikiHow, the demonstration learning example here is selected by random retrieval from the demo list, which is entirely unrelated with the theme of test instructional text.

Part	Two-shot Random-retrieval Prompt Text
Demo1	[Message]: How to Use GIMP: 1. Download the latest version of GIMP (GNU Image Manipulation Program): You can find it for free from the developer's website here. Click the Download GIMP X.X.X link under the GIMP for Windows heading. The setup file will begin downloading in a few seconds. 2. Run the installer: Windows will ask if you want to run the file. Double-check to make sure that you downloaded GIMP from the developer. Select your language to proceed with the installation.
	[Summarisation]: How to Use GIMP: 1. Install the latest version of GIMP. 2. Run the installer. 3. Open the app. 4. Click File > New to create a new image. 5. Click File > Open to open an existing image. 6. Crop, flip, and rotate in the Tools menu. 7. Add text, lines, and illustrations using the toolbar.
Demo2	[Message]: How to Allow 3rd Party Cookies on a Mac: 1. Open Safari: Click the Safari app icon, which resembles a blue compass, in your Mac's Dock. 2. Click Safari: It's a menu item in the top-left corner of the screen next to the Apple icon. 3. Click Preferences…: This option is near the top of the drop-down menu. 4. Click the Privacy tab: It's at the top of the Preferences window. 5. Deselect the "Block All Cookies" option: This option is below the "Cookies and website data" heading. Doing so will enable cookies in your Safari browser.
	[Summarisation]: How to Allow 3rd Party Cookies on a Mac: 1. Open Safari. 2. Click Safari. 3. Click Preferences. 4. Click the Privacy tab. 5. Deselect the "Block All Cookies" option.
Ask	[Message]: How to Use the Red Eye Tool in Photoshop: 1. Put the image that you want up in Photoshop. While editing, you want it as large as you can get it.; 2. In the toolbox, right click on the plaster (band-aid). Click on the icon that looks like an eye and is labeled "Red eye tool". 3. Click the eyes on your image in order to remove the red eye. You will know when you have done this successfully because you will notice the eyes are no longer flashy and the image looks much better. 4. You have now removed the redeye and you can now do what you want with the image, get it framed, put it on Facebook, etc. Be sure to downsize it when you are finished.
	[Summarisation]: ?

Table 10: Two-shot in-context learning prompting example to generate the instructional summary answer of **How to Use the Red Eye Tool in Photoshop** from WikiHow, the demonstration learning examples here are selected by random retrieval from the demo list, which are entirely unrelated with the theme of test instructional text.

Part	Two-shot Similarity-retrieval Prompt Text
Demo1	[Message]: How to Rotate an Image in Photoshop: 1. Open your image in Photoshop: If you'd like to rotate or flip an entire image, click "File," then "Open." Select the image you wish to rotate and click "Open" once more. 2. Select a rotation option: Navigate to Image >> Image Rotation to view several options for rotation. 3. Undo your changes: If you're not happy with the flip or rotate option you've selected, press Ctrl+Z (Windows) or Command+Z (Mac) to undo the action. 4. Save the image: Open the File menu and click "Save As." Choose a location to which you'll save your newly rotated image.
	[Summarisation]: How to Rotate an Image in Photoshop: 1. Click the Image menu. 2. Select Image Rotation. 3. Choose a degree and direction. 4. Press Ctrl + Z or Cmd + Z to undo if needed.
Demo2	[Message]: How to Cut an Image on Photoshop: 1. Open your project in Photoshop: You can open an image in Photoshop by right-clicking the image file, selecting Open with… and Photoshop. 2. Click the Quick Selection tool: This is in the menu to the left in the application window. You can also press W. The Quick Selection tool selects all pixels that are similar in color. 3. Click and drag on what you want to delete: The Quick Selection tool will select similar pixels to what you clicked on. 4. Refine the edges of your selection: You'll find the option to Refine Edge in the Select tab in the menu you can find either along the top of your screen or the top of the application window. You'll see your image with the image cut previewed. 5. Press Ctrl+X or Cmd+X to cut your selection: Your selection will disappear from the canvas, but it is copied in your clipboard.
	[Summarisation]: How to Cut an Image on Photoshop: 1. Open your project in Photoshop. 2. Click the Quick Selection tool. 3. Click on your object to cut. 4. Refine your cut selection. 5. Press Ctrl+X or Cmd+X to cut your selection.
Ask	[Message]: How to Use the Red Eye Tool in Photoshop: 1. Put the image that you want up in Photoshop. While editing, you want it as large as you can get it.; 2. In the toolbox, right click on the plaster (band-aid). Click on the icon that looks like an eye and is labeled "Red eye tool". 3. Click the eyes on your image in order to remove the red eye. You will know when you have done this successfully because you will notice the eyes are no longer flashy and the image looks much better. 4. You have now removed the redeye and you can now do what you want with the image, get it framed, put it on Facebook, etc. Be sure to downsize it when you are finished.
	[Summarisation]: ?

Table 11: Two-shot in-context learning prompting example to generate the instructional summary answer of **How to Use the Red Eye Tool in Photoshop** from WikiHow, the demonstration learning examples here are selected by similarity retrieval from the demo list. The examples selected only focus on the 'photoshop', which loses the generality.

Part	**Two-shot Auto-prompt Prompt Text**
Demo1	[Message]: How to Rotate an Image in Photoshop: 1. Open your image in Photoshop: If you'd like to rotate or flip an entire image, click "File," then "Open." Select the image you wish to rotate and click "Open" once more. 2. Select a rotation option: Navigate to Image >> Image Rotation to view several options for rotation. 3. Undo your changes: If you're not happy with the flip or rotate option you've selected, press Ctrl+Z (Windows) or Command+Z (Mac) to undo the action. 4. Save the image: Open the File menu and click "Save As." Choose a location to which you'll save your newly rotated image.
	[Summarisation]: How to Rotate an Image in Photoshop: 1. Click the Image menu. 2. Select Image Rotation. 3. Choose a degree and direction. 4. Press Ctrl + Z or Cmd + Z to undo if needed.
Demo2	[Message]: How to Remove Red Eye on iPhone, iPod, and iPad Photos: 1. Open the Photos app: It's the white icon with a multicolored flower. You will find it on your home screen or in the "Creativity" folder of your App Library. 2. Tap the photo you want to edit: To view all of your photos, you can tap the Albums tab at the bottom and then select All Photos. When you tap a photo, it will open in Photos. 3. Tap Edit: It's at the top-right corner. 4. Tap the "Red Eye Correction" icon: It's the icon of an eye with a line through it, and you'll find it at the top-right corner of the screen. 5. Tap each red eye: Red Eye Correction will automatically alter the pixels in the areas you tap. 6. Tap Done: It's at the bottom-right corner of the screen. This saves your changes.
	[Summarisation]: How to Remove Red Eye on iPhone, iPod, and iPad Photos: 1. Open the Photos app. 2. Tap the photo. 3. Tap Edit. 4. Tap the eyeball icon. 5. Tap each red eye. 6. Tap Done.
Ask	[Message]: How to Use the Red Eye Tool in Photoshop: 1. Put the image that you want up in Photoshop. While editing, you want it as large as you can get it.; 2. In the toolbox, right click on the plaster (band-aid). Click on the icon that looks like an eye and is labeled "Red eye tool". 3. Click the eyes on your image in order to remove the red eye. You will know when you have done this successfully because you will notice the eyes are no longer flashy and the image looks much better. 4. You have now removed the redeye and you can now do what you want with the image, get it framed, put it on Facebook, etc. Be sure to downsize it when you are finished.
	[Summarisation]: ?

Table 12: Two-shot in-context learning prompting example to generate the instructional summary answer of **How to Use the Red Eye Tool in Photoshop** from WikiHow, the demonstration learning examples here are selected by autoprompt retrieval from the demo list. The examples selected not only focus on 'photoshop' but also 'Red Eye', which achieves a good balance between diversity and similarity.

Part	Two-shot Auto-prompt + CoT Table Prompt Text
Demo1	[Message]: How to Rotate an Image in Photoshop: 1. Open your image in Photoshop: If you'd like to rotate or flip an entire image, click "File," then "Open." Select the image you wish to rotate and click "Open" once more. 2. Select a rotation option: Navigate to Image >> Image Rotation to view several options for rotation. 3. Undo your changes: If you're not happy with the flip or rotate option you've selected, press Ctrl+Z (Windows) or Command+Z (Mac) to undo the action. 4. Save the image: Open the File menu and click "Save As." Choose a location to which you'll save your newly rotated image.
	[How to Summarisation]: Col 1 Action — Col 2 Result — Col 3 Purpose — Col 4 Relation Open image — Image opened — Rotate image — Parallel Select option — Option chosen— Rotate image — Causal Undo changes — Changes undone — Undo action — Causal Save image — Image saved — Save changes — Causal
	[Summarisation]: How to Rotate an Image in Photoshop: 1. Click the Image menu. 2. Select Image Rotation. 3. Choose a degree and direction. 4. Press Ctrl + Z or Cmd + Z to undo if needed.
Demo2	[Message]: How to Remove Red Eye on iPhone, iPod, and iPad Photos: 1. Open the Photos app: It's the white icon with a multicolored flower. You will find it on your home screen or in the "Creativity" folder of your App Library. 2. Tap the photo you want to edit: To view all of your photos, you can tap the Albums tab at the bottom and then select All Photos. When you tap a photo, it will open in Photos. 3. Tap Edit: It's at the top-right corner. 4. Tap the "Red Eye Correction" icon: It's the icon of an eye with a line through it, and you'll find it at the top-right corner of the screen. 5. Tap each red eye: Red Eye Correction will automatically alter the pixels in the areas you tap. 6. Tap Done: It's at the bottom-right corner of the screen. This saves your changes.
	[How to Summarisation]: Col 1 Action — Col 2 Result — Col 3 Purpose — Col 4 Relation Open Photos — Photos opened — Remove red eye — Parallel Tap Photo — Photo opened — Edit photo — Causal Tap Edit — Edit menu open— Edit photo — Causal Tap "Red Eye Correction" — Red eye corrected — Remove red eye — Causal Tap each red eye — Red eye corrected — Remove red eye — Causal Tap Done — Changes saved — Save changes — Causal
	[Summarisation]: How to Remove Red Eye on iPhone, iPod, and iPad Photos: 1. Open the Photos app. 2. Tap the photo. 3. Tap Edit. 4. Tap the eyeball icon. 5. Tap each red eye. 6. Tap Done.
Ask	[Message]: How to Use the Red Eye Tool in Photoshop: 1. Put the image that you want up in Photoshop. While editing, you want it as large as you can get it.; 2. In the toolbox, right click on the plaster (band-aid). Click on the icon that looks like an eye and is labeled "Red eye tool". 3. Click the eyes on your image in order to remove the red eye. You will know when you have done this successfully because you will notice the eyes are no longer flashy and the image looks much better. 4. You have now removed the redeye and you can now do what you want with the image, get it framed, put it on Facebook, etc. Be sure to downsize it when you are finished.
	[Summarisation]: ?

Table 13: Two-shot in-context learning prompting example to generate the instructional summary answer of **How to Use the Red Eye Tool in Photoshop** from WikiHow, the demonstration learning examples here are selected by autoprompt retrieval + CoT table from the demo list. The added section is the so-called CoT table that lists the action, purpose, result, and relationship with the last step information for each instruction. This component will enhance the reasoning ability of ICL in LLMs.

Prompting	Generated Instructional Summary from T5-Base Model
One-shot Random-retrieval	[Summarisation]: How to Use the Red Eye Tool in Photoshop: 1. Put the image that you want up in Photoshop. 2. In the toolbox, right-click on the plaster (band-aid). Click on the icon that looks like an eye and is labeled "Red eye tool". 3. Click the eyes on your image in order to remove the red eye. 4. You will know when you have done this successfully because you will notice the eyes are no longer flashy and the image looks much better.
Two-shot Random-retrieval	[Summarisation]: How to Use the Red Eye Tool in Photoshop: 1. Put the image that you want up in Photoshop. 2. In the toolbox, right click on the plaster (band-aid). Click on the icon that looks like an eye and is labeled "Red eye tool". 3. Click the eyes on your image in order to remove the red eye. 4. You will know when you have done this successfully because you will notice the eyes are no longer flashy and the image looks much better. 5. You have now removed the redeye and you can now do what you want with the image
Two-shot Similarity-retrieval	[Summarisation] How to Use the Red Eye Tool in Photoshop: 1. Put the image that you want up in Photoshop. 2. In the toolbox, right click on the plaster (band-aid). Click on the icon that looks like an eye and is labeled "Red eye tool". 3. Click the eyes on your image to remove the red eye. 4. You have now removed the redeye and you can now do what you want with the image, get it framed, put it on Facebook, etc. Be sure to downsize it when you are finished.
Two-shot Autoprompt	[Summarisation]: How to Use the Red Eye Tool in Photoshop: 1. Open the image you want to edit. 2. In the toolbox, right click on the plaster (band-aid). Click on the icon that looks like an eye and is labeled "Red eye tool". 3. Click the eyes on your image in order to remove the red eye. 4. You will know when you have done this successfully because you will notice the eyes are no longer flashy and the image looks much better. 5. You have now removed the redeye and you can now do what you want with the image, get it framed
Two-shot Autoprompt + CoT	[Summarisation]: How to Use the Red Eye Tool in Photoshop: 1. Put the image up in Photoshop. 2. Right click on the plaster and select the Red Eye Tool. 3. Click the eyes to remove the red eye. 4. Downsize the image when finished.

Table 14: The generated ICL instructional summary based on different prompting engineering methods on **How to Use the Red Eye Tool in Photoshop** from WikiHow.