Exploring Memorization in Fine-tuned Language Models

Shenglai Zeng[1*†], **Yaxin Li**[1*], **Jie Ren**[1], **Yiding Liu**[2], **Han Xu**[1], **Pengfei He**[1]
Yue Xing[1], **Shuaiqiang Wang**[2], **Jiliang Tang**[1], **Dawei Yin**[2]
[1]Michigan State University [2]Baidu, Inc.
{zengshe1, liyaxin1, renjie3, xuhan1, hepengf1,xingyue1, tangjili}@msu.edu,
liuyiding.tanh@gmail.com, shqiang.wang@gmail.com, yindawei@acm.org

Abstract

Large language models (LLMs) have shown great capabilities in various tasks but also exhibited memorization of training data, raising tremendous privacy and copyright concerns. While prior works have studied memorization during pre-training, the exploration of memorization during fine-tuning is rather limited. Compared to pre-training, fine-tuning typically involves more sensitive data and diverse objectives, thus may bring distinct privacy risks and unique memorization behaviors. In this work, we conduct the first comprehensive analysis to explore language models' (LMs) memorization during fine-tuning across tasks. Our studies with open-sourced and our own fine-tuned LMs across various tasks indicate that memorization presents a strong disparity among different fine-tuning tasks. We provide an intuitive explanation of this task disparity via sparse coding theory and unveil a strong correlation between memorization and attention score distribution.

1 Introduction

Large language models (LLMs) have demonstrated impressive capabilities in natural language understanding and generation, enabling significant advances across diverse applications including reading comprehension, text classification, and summarization (OpenAI, 2023; Ouyang et al., 2022; Bai et al., 2022; Touvron et al., 2023b). However, recent works reveal that pre-trained langauge models (LMs) tend to memorize and regenerate segments of their pre-training corpus when prompted appropriately. For example, Carlini et al. (2021) devised a training data extraction attack, successfully extracting hundreds of verbatim text sequences from GPT-2's training data. Existing works demonstrate that various factors can affect memorization and memorization effects grow with model scale, data duplication, and prompt length (Lee et al., 2021; Kandpal et al., 2022; Carlini et al., 2022). These findings raise privacy and confidentiality concerns, as interactions between humans and the deployed LMs could enable extraction of the memorized sensitive training data, such as phone numbers, people's names, etc. As the scale of LMs and their training data continues to expand, the privacy risks posed by memorization become increasingly serious.

In addition to pre-training, the application of LMs often involves fine-tuning on downstream tasks (Touvron et al., 2023b; Chung et al., 2022; Ouyang et al., 2022; Longpre et al., 2023), while the memorization of fine-tuning data is rather overlooked by existing studies. Compared to pre-training, fine-tuning introduces two unique perspectives with respect to memorization. **First, fine-tuning often utilizes domain-specific and private data.** For instance, developing a diagnostic chatbot (Yunxiang et al., 2023) requires collecting sensitive medical conversation data. Similarly, an academic LM (Beltagy et al., 2019) may be trained on copyrighted essays for summarization or paraphrase generation. Leakage of such fine-tuning data can seriously violate user privacy or intellectual property rights(Ren et al., 2024). **Second, fine-tuning involves more complex and diverse training goals compared to pre-training.** During pre-training, the learning objective is usually language modeling from a massive unlabeled corpus (e.g., next-word prediction), which is agnostic to downstream tasks. In fine-tuning, the objective is to learn task-specific knowledge from annotated data, such as how to effectively capture the key information of a long document for summarization. The differences may induce distinct memorization behaviors and patterns during fine-tuning. Consequently, it is necessary to explore memorization for fine-tuning. Yet, it is challenging because previous insights and findings regarding the pre-trained models may not directly apply to fine-tuning.

[*]Equal contribution.
[†]Work conducted during an internship at Baidu, Inc.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3917–3948

To bridge this gap, we focus on the memorization of LMs during fine-tuning. We study a variety of fine-tuning tasks including summarization, dialogue, question answering, machine translation, and sentiment analysis. Using an automatic plagiarism detection pipeline (Lee et al., 2023), we examine memorization on both popular open-sourced models and the models fine-tuned for diverse tasks. In both cases, we consistently observe the existence of substantial memorization under certain tasks. Moreover, we draw several new insights and reveal potential factors that may impact the memorization of fine-tuned LMs. Our key findings and contributions are summarized as follows:

- *Disparate Memorization Across Tasks.* Particular tasks such as summarization and dialogue present high memorization. In contrast, tasks like classification, reading comprehension, and translation exhibit low memorization. This discrepancy highlights the disparate cognitive demands these tasks need from LMs.
- *Task-Dependent Scaling in Fine-tuned Memorization.* For tasks with high memorization, the degree of memorization escalates with the increase of model size. On the other hand, for tasks with low memorization, increasing the model size does not significantly amplify the memorization.
- *Memorization Linked to Task Information Needs.* We hypothesize that the varying degrees of memorization across different tasks are linked to the number of input features that LMs need to retain. We further justify this based on sparse coding models and attention patterns. Specifically, tasks which need to understand every detail of the input tends to memorize more from the data, and the distribution of the attention score is dense across all input-output pairs.

2 Related Work

Powered by the transformer architecture (Vaswani et al., 2017), in recent years, LMs such as ChatGPT (Ouyang et al., 2022), Claude (Bai et al., 2022), Palm (Chowdhery et al., 2022), Llama (Touvron et al., 2023b,a) and T5 (Raffel et al., 2020) have achieved impressive performance across a wide range of natural language processing (NLP) tasks. These language models are pre-trained by a large amount of data to enhance their overall proficiency. Subsequently, people usually utilize various techniques (Chung et al., 2022; Ouyang et al., 2022; Houlsby et al., 2019; Hu et al., 2021; Li and Liang, 2021; Lester et al., 2021; Liu et al., 2021) to fine-tune the pre-trained models, thus enabling them to more effectively adapt to different downstream tasks.

The memorization behavior of pre-trained LMs has attracted increasing attention in recent years. Carlini et al. (2021) first proposed a data extraction attack, demonstrating that LMs tend to memorize and regenerate segments of training data. Kandpal et al. (2022) and Lee et al. (2021) revealed that duplicated training data is more vulnerable to memorization, and de-duplication can effectively reduce memorization. Carlini et al. (2022) further quantified memorization effects, revealing that memorization grows with model scale, data duplication, and prompt length.

There are also works providing different views and understandings on memorization. For example, Ippolito et al. (2022) developed an efficient defense preventing memorizing the exact sentences (verbatim memorization), yet showed it fails to prevent leakage of training data. This shows the need for definitions beyond verbatim memorization. To distinguish "common" memorization from "rare" memorization, Zhang et al. (2021) formulated a new notion of counterfactual memorization, which measures how predictions change if a particular document is deleted during training. Biderman et al. (2023) investigated predictable memorization by extrapolating small or partially-trained LMs' behavior to forecast memorization in larger models. They further presented scaling laws of prediction and explored ways to improve prediction reliability.

While most literature focused on memorization during pre-training, limited work has investigated memorization in the fine-tuning stage. Mireshghallah et al. (2022) examined memorization risks in different fine-tuning methods for large LMs. They found that fine-tuning only the head leads to higher memorization compared to fine-tuning smaller adapter modules. Lee et al. (2023) studied plagiarism during fine-tuning, concluding that the plagiarism patterns in fine-tuned LMs depend on corpus similarity and homogeneity. However, these studies considered fine-tuning with the same objective as pre-training, which is different from the the common practice of fine-tuning in various tasks. As a result, in this paper, we focus on the more general and realistic scenario of multifaceted fine-tuning across diverse objectives.

3 Preliminary

In this section, we first introduce the definition of memorization and the detection methods used in this paper, and then introduce our preliminary findings on open-sourced fine-tuned LMs across various tasks.

3.1 Definitions and Notations

Definitions of memorization in literature. In literature, there are some definitions of memorization. For example, in Carlini et al. (2022), a straightforward and strict definition is that a string s is extractable with its context p (with length k) if the concatenation $[p\|s]$ exists in the training set and $f(p)$ produces exactly the output of s, i.e., $f(p) = s$. This is defined as **verbatim memorization**. In Ippolito et al. (2022), a relaxed definition of memorization is using the Bilingual Evaluation Understudy (BLEU) score.

However, the above two definitions only consider memorizing the exact wordings of the data, instead of memorizing the meaning of the data. Therefore, in Lee et al. (2023), plagiarism detection tools are leveraged to to identify memorization through comparing the machine-generated text with the whole training set .

Definition of fine-tuned memorization. In the fine-tuning stage, models are trained for specific tasks like sentiment analysis, dialog, and summarization. We define fine-tuning as supervised training using samples $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{n}$, where y_i represents the target output for input x_i. Since the input texts usually contain more information than the output in our considered tasks, we majorly discuss the potential information leakage from the input corpus in the training set, i.e., $\mathcal{D}_{\text{input}} = \{x_i\}_{i=1}^{n}$.

To explore memorization, we follow the prompting approach (Carlini et al., 2022) by dividing each $x_i = [p_i\|s_i]$ to a length-k prefix p_i, and a suffix s_i. We further define the set of all prefixes in the training set as $P = \{p_i\}_{i=1}^{n}$, and the set of all suffixes as $S = \{s_i\}_{i=1}^{n}$. We define fine-tuned memorization as follows:

Definition 1 (Fine-tuned memorization) *Given the fine-tuned model function f, fine-tuned memorization is defined as when the model output $f(p_i)$ contains information of any $s_j \in S$, formalized by $D(f(p_i), s_j) = True$, where D is a discriminative function to judge the similarity between two texts.*

Evaluation method. In our practice, we input $P = \{p_i\}_{i=1}^{n}$ to the model and utilize a local search engine to quickly locate suspicious texts and use a plagiarism detection tool to serve as the discriminative function D. In detail, given a dataset of size n with suffix space S, we employ the local search engine, Elasticsearch[1], to identify the top-K corpus candidates $S_K^i = \{s_1^i, s_2^i, \ldots, s_K^i\}$ which are similar to $f(p_i)$. Then we utilize the PAN2014 plagiarism detection tool[2] to assess the similarity between $f(p_i)$ and each candidate $s_j \in S_k^i$. This detection tool is capable of identifying the presence of plagiarised pairs (d_i, d_j), where d_i and d_j are sub-strings from $f(p_i)$ and s_j, respectively. The main idea of this tool is to transform text into term frequency-inverse document frequency (TF-IDF) vectors and utilize sentence similarity measures (cosine similarity) to identify plagiarism cases. We say the fine-tuned model memorizes s_j if the plagiarism is confirmed. We then count the number of memorized cases and divide by n to get the total memorization rate. We use $n = 10000$ in our experiments. This memorization rate quantifies the memorization exhibited in the model.

Moreover, the detection tool can categorize memorized content into three distinct types. To provide a detailed quantification of the memorization behavior, we include these categories following the methods in prior work (Lee et al., 2023).

- **Verbatim**: d_j is an exact replica of d_i.
- **Paraphrase**[3]: d_j is a rephrased version of d_i
- **Idea memorization**: d_j condenses d_i into fewer sentences, or vice versa.

Generally, all these memorization types indicate that the model generates information about the suffix of training data x not given as input. More details of the detection pipeline, e.g., implementation descriptions and differences among memorization types, are included in the Appendix B. Typical memorization cases are shown in Appendix J.

3.2 Preliminary Findings

To initially explore the memorization effects during fine-tuning, we examine several popular open-sourced fine-tuned models from HuggingFace that were fine-tuned on 6 representative tasks.We attach the descriptions of all models and datasets

[1] https://www.elastic.co/elasticsearch/

[2] https://pan.webis.de/clef14/pan14-web/text-alignment.html

[3] Paraphrasing is further assessed using RoBERTa and NER, classifying $p < 0.5$ as low-confidence and $p > 0.5$ as high-confidence, with both reported.

Table 1: Memorization rate of open-sourced LLMs fine-tuned on various tasks

Task	Dataset	Source Model	Total Mem Rate	Verbatim	Idea	Paraphrase ($p > 0.5$)	Paraphrase ($p < 0.5$)
Summarization	CNN/Daily Mail	Bart_Large	20.7%	1.3%	0%	9.8%	9.6%
Medical Dialog	ChatDoctor	BioGPT	19.6%	0.1%	3.5%	7.8%	8.2%
Extractive QA	SQuAD_v2	T5_large	0.1%	0%	0%	0%	0.1%
Abstractive QA	Race	T5_large	0.3%	0%	0%	0.2%	0.1%
Translation	WMT_19	FSMT	0%	0%	0%	0%	0%
Sentiment Classification	IMDB	T5-base	0%	0%	0%	0%	0%

used in Appendix H. These tasks include summarization, medical dialog, question and answering (QA), translation, and sentiment analysis. The preliminary results in Table 1 suggest that substantial memorization of the fine-tuning data occurs in fine-tuning. For summarization and medical dialog models, we identified total memorization rate of 20.7% and 19.6%, respectively. These high rates could imply potential privacy violations or copyright issues. Furthermore, the level of memorization varies across tasks. Models fine-tuned for summarization and medical dialog exhibit high memorization, while models for remaining tasks show much lower memorization. These observations motivate further in-depth analysis to validate the observed task-specific memorization behavior in Section 4 and the possible scaling effect in Section 5 to fine-tune models by ourselves. Furthermore, in Section 6, we provide an in-depth analysis and understanding of the potential reason behind the observed disparate fine-tuned memorization across tasks.

4 Disparate Memorization Across Tasks

Our preliminary study demonstrates that the memorization of fine-tuned models varies on different fine-tuning tasks. However, the causes of such difference are still unclear, which could be fine-tuning datasets or model architectures. To clarify this, in Section 4.1, we control the impact of different variables to more precisely explore the relation between fine-tuning task and the memorization effect. In Section 4.2, we further examine the impact of decoding methods and prefix length. Note that in our fine-tuning process, we ensure that our fine-tuned models have satisfactory performance on downstream tasks, see Appendix I.

4.1 Fine-tuned with Fixed Pre-trained LM and Dataset

Fine-tuned on T5-base LM. To eliminate the potential impact from the pre-trained LM, we conduct experiments to fine-tune the same pre-trained T5-base model[4] for different fine-tuning tasks. We treat all the tasks as generative tasks (like instruction-tuning) and do not add any additional modules (e.g., MLP). Details of the datasets and memorization results are presented in Table 2[5]. In case a part of the fine-tuning data appears in the pre-training data, we report the memorization rate of the pre-trained model[6] and present the change in the memorization rate before and after fine-tuning. Table 2 clearly shows a substantial total memorization rate and memorization gain for summarization tasks (22.3%, ↑8.35%) and medical dialogue (8.27%, ↑6.67%). Meanwhile, the memorization and the gain in fine-tuning are much lower for reading comprehension (0.15%, ↑0.07%), translation (0.0%, ↑0.0%), and sentiment classification (0.8%, ↑0.02%). These observations are consistent with our preliminary findings on open-sourced models and suggest that fine-tuned memorization with the same pre-trained LM architecture still demonstrates a strong task disparity.

Fine-tuned on different tasks with the same dataset. In this experiment, we investigate the memorization of different tasks fine-tuned on the same pre-trained LM with the same dataset. Specifically, we fine-tune the T5-base model on Rent-TheRunway dataset (Misra et al., 2018a), which contains self-reported clothing fit feedback from customers along with additional metadata. Each product has multiple attributes, including customer reviews, ratings, review summaries, review dates, etc. It allows us to fine-tune the same pre-trained model (i.e., T5-base), with the identical inputs (i.e.,

[4]We also fine-tuned GPT-Neo models and found consistent findings with T5, we report the results in Appendix C and Table 9. Besides, memorization results on other tasks and datasets and memorization behavior of multi-task fine-tuned models are reported in Appendix D and E

[5]We provide the statistical significance testing of memorization rate in Appendix G.

[6]In the context of pre-trained models, we continue to utilize the prefixes from the fine-tuning dataset for evaluating memorization. This process is detailed in Appendix B.1.

Table 2: Memorization rate of T5-base fine-tuned on various tasks

Task	Dataset	Model	Total Mem Rate	Verbatim	Idea	Paraphrase (P>0.5)	Paraphrase (P<0.5)
Summarization	Multi-news	T5-base	13.98%	3.58%	0.66%	4.28%	5.46%
		T5-finetuned	22.33%	4.23%	0.65%	6.23%	11.22%
		Difference	↑8.35%	↑0.65%	↓0.01%	↑1.95%	↑5.76%
Dialog	chatdoctor	T5-base	1.60%	0.03%	1.04%	0.11%	0.42%
		T5-finetuned	8.27%	0.02%	1.41%	1.75%	5.09%
		Difference	↑6.67%	↓0.01%	↑0.37%	↑1.64%	↑4.67%
Sentiment Classification	imdb	T5-base	0.78%	0.05%	0.37%	0.16%	0.20%
		T5-finetuned	0.80%	0.04%	0.30%	0.17%	0.29%
		Difference	↑0.02%	↓0.01%	↓0.07%	↑0.01%	↑0.09%
Reading Comprehension	Squad_v2	T5-base	0.08%	0.02%	0.00%	0.01%	0.05%
		T5-finetuned	0.15%	0.04%	0.00%	0.05%	0.06%
		Difference	↑0.07%	↑0.02%	-	↑0.04%	↑0.01%
Translation	wmt	T5-base	0.00%	0.00%	0.00%	0.00%	0.00%
		T5-finetuned	0.00%	0.00%	0.00%	0.00%	0.00%
		Difference	-	-	-	-	-

the customer reviews), for different task objectives (i.e., review summarization and sentiment classification). Note that we map the ratings into positive and negative labels for fine-tuning sentiment classification model. The memorization performance of these two models is shown in Figure 1. The results demonstrate that the summarization model exhibits higher memorization compared to the classification model, which validates that the task objective impacts the memorization of fine-tuned models.

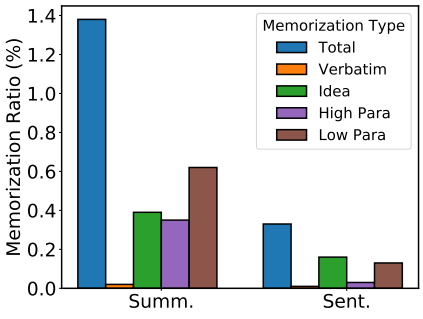


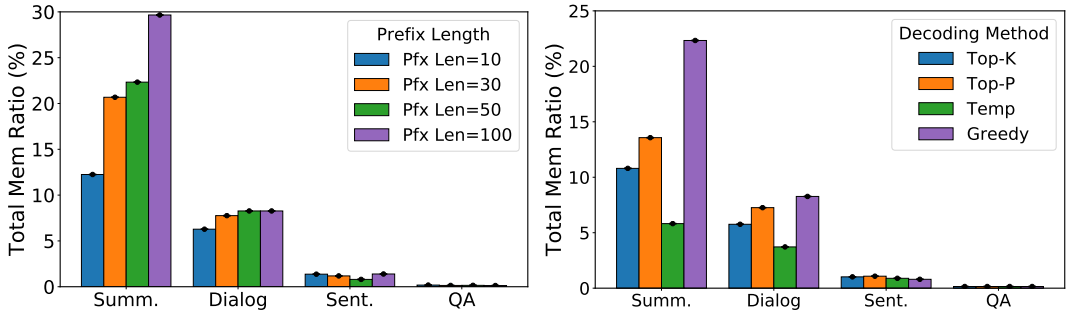
Figure 1: Memorization of T5-base fine-tuned on Rent-TheRunway.

4.2 Further Probing

The memorization behavior of pre-trained models has been shown to be influenced by factors such as generation sampling methods and input prefix lengths (Carlini et al., 2022; Lee et al., 2023). In this subsection, we extend our investigation to explore how these factors affect memorization in fine-tuned models. We also conduct an ablation study on the impact of training epochs on memorization and report the results in Appendix F.1 and Table 5.

Input prefix length. We vary the length of the prefix of inputs and report the main results in Figure 2a. We find that the length of prefix tokens influences memorization differently across tasks. In summarization and dialogue tasks, the total memorization rate tends to increase with longer prefixes, aligning with existing research on pre-trained memorization. However, in sentiment classification and QA, altering the prefix length does not significantly affect memorization. More results in Appendix F.3 and Table 7. **Despite these variations, a consistent disparity in memorization across different tasks persists, regardless of the prefix length.**

Generation sampling. We study the impact of different generation sampling methods including (i) top-k (k=40) sampling, (ii) top-p (p=0.8) sampling and (iii) changing the temperature to T=1, and report the main results in Figure 2b and more results in Appendix F.2 and Table 6. It is observed that sampling affects memorization differently across tasks: it lowers memorization in high-memorization tasks like summarization and dialogue, but has a negligible or even increasing effect on memorization in low-memorization tasks such as sentiment classification. **Despite these variations, a significant, consistent disparity in memorization remains across different tasks, indicating an intrinsic, task-specific inclination towards memorization that is not significantly altered by sampling methods.**



(a) Prefix length

(b) Sampling methods

Figure 2: Impact of prefix length and sampling methods on memorization

5 Memorization Scaling Behavior of Fine-tuned LMs

It is evident that memorization in pre-trained models tends to increase with model size. To understand the scaling behavior of memorization in fine-tuned models, we conduct a systematic analysis on fine-tuned models, comparing the memorization in various tasks using different sizes of the T5 model: T5-small (60M), T5-base (220M), T5-large (770M), and T5-xl (3B).

5.1 High-memorization Tasks

We explore the scaling behavior in high-memorization tasks, specifically focusing on summarization (fine-tuned on Multi-news) and dialogue (fine-tuned on Chatdoctor). The results can be found in Figures 3a and 3b respectively. We can see that memorization in fine-tuning increases with model size. As a benchmark, we also provide the memorization rate for the pre-trained model, for which we can see that when increasing from a model size of 220M, the memorization rate does not further increase much. These two observations in the fine-tuned model and the pre-trained model together reveal that the fine-tuned model is memorizing information from the fine-tuning data, indicating severe privacy threats when scaling up the models in these tasks.

5.2 Low-memorization Tasks

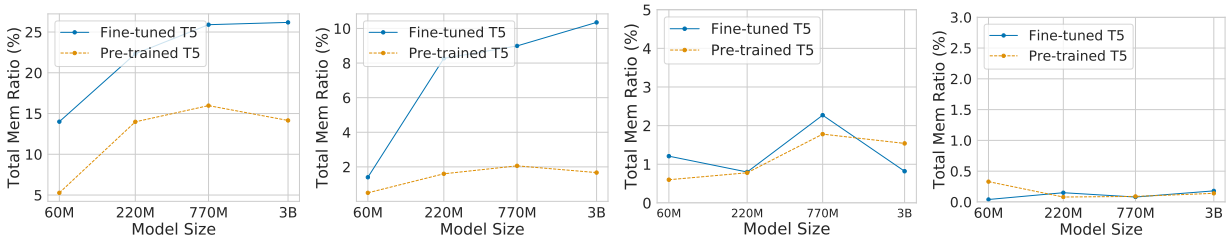
In contrast to high-memorization tasks, low-memorization tasks such as sentiment classification and question answering exhibit different scaling behaviors. As illustrated in Figures 3c and 3d, an increase in model size does not result in a rise in the memorization rate, and the memorization rate is consistently low. This suggests that even when large models are fine-tuned on these tasks, the possibility of memorization and outputting fine-tuning data is relatively low.

6 Understanding the Memorization Disparity

In the previous sections, we empirically examine the memorization rate and scaling behavior among a variety of tasks, and demonstrate the discrepancy between high- and low-memorization tasks. In this section, we provide understanding and evidence on the underlying reason behind such disparity of fine-tuned memorization.

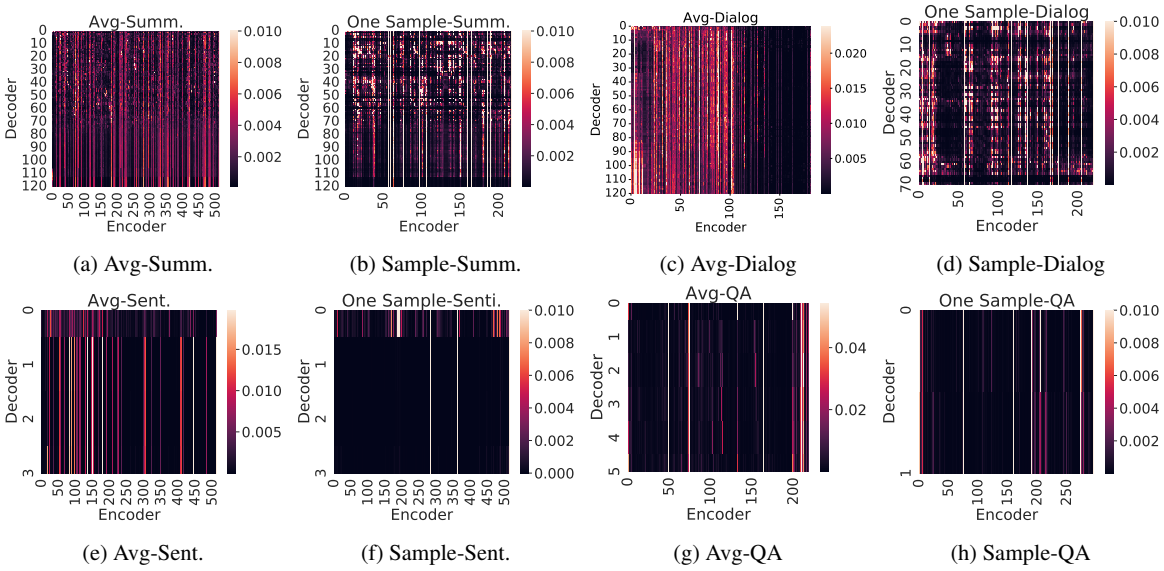
6.1 Correlation between Memorization and Task-specific Information

In this subsection, we aim to investigate the question: *why do different fine-tuning tasks present different memorization behaviors?* We conjecture that the memorization behavior might be closely related to the information needed to fulfill certain language tasks. Intuitively, for language tasks such as sentiment analysis or extractive QA, only a few words or sentences are enough for the model to complete the task. For example, one can determine the sentiment based on some specific words in the sentiment, and can answer a question based on certain pieces of information. In this case, the model only needs to learn specific key features and is less likely to memorize the other data. On the other hand, for tasks such as summarization and dialog, they require the model to learn more input features to complete the task, as the essential information from these inputs is also reflected in the output. As a result, the fine-tuning process will encode more input knowledge from the data in the model parameters, leading to potential concerns of memorization. In the following, we provide a conceptual discussion based on the sparse decoding model. The sparse coding model is a method that represents original data by focusing on its key features, using only the most crucial elements to efficiently express the core information, which is a popular model for modeling



(a) Summarization (b) Dialog (c) Sentiment classification (d) Extractive QA

Figure 3: Scaling behavior of fine-tuned memorization



(a) Avg-Summ. (b) Sample-Summ. (c) Avg-Dialog (d) Sample-Dialog

(e) Avg-Sent. (f) Sample-Sent. (g) Avg-QA (h) Sample-QA

Figure 4: Decoder-encoder attention heatmaps for (a, b) Summarization, (c, d) Dialog, (e, f) Sentiment Analysis, and (g, h) QA. (a, c, e, g) show average heatmaps from 10 samples, while (b, d, f, h) show heatmaps from a single sample.

text and vision data. (Arora et al., 2015, 2018; Olshausen and Field, 1997, 2004)

Sparse coding model. Denote an observed text data as $Z \in \mathbb{R}^{d \times D}$ where D and d represent the sequence length and the length of the embedding respectively. The basic assumption of sparse coding is that the data Z comes from combinations of a few hidden features. We use X to represent these hidden features and consider the following relation:

$$Z = UXV \tag{1}$$

where $X \in R^{k \times K}$, $U \in \mathbb{R}^{d \times k}$, $V \in \mathbb{R}^{K \times D}$, $k \leq d$, $K \leq D$. Each column of U is a unit vector and orthogonal with each other. Each row of V is a unit vector and orthogonal with each other. Given the above formulation, each element in Z is a linear combination of the elements from X. Compared to previous literature, our assumption is a simplified version of the original sparse coding model as we do not further impose noise in the data generation model. Besides, we also modify the original 1D feature X in sparse coding into 2D for our model.

Task complexity. Under the sparse coding model, we further assume that the output can be fully expressed by a linear transformation of X. However, different fine-tuning tasks may differ in *how much input information is needed by the task*. We present two perspectives below to illustrate why "complex tasks" may have more memorization.

First, the number of parameters to connect Z with the final target is related to the task-specific information. Consider we have a *"simple task"* (e.g., sentiment analysis) where the model output is only one scalar (preference) decided by some certain features or a combination of X. In the simplest case, the scalar is a linear function of X, $a^\top Xb$, where $a \in \mathbb{R}^k$ and $b \in \mathbb{R}^K$. In this case, the loss function for a sentiment classification model f_{cls} can be defined as:

$$l(f_{cls}(Z), a^\top Xb). \tag{2}$$

If we further assume the loss functions as square loss, the best solution of f_{cls} is: $f_{cls}(Z) = (a')^\top Z b'$ where $a^\top U^\top := a'$ and $V^\top b := b'$. It means that the model only needs to learn two vector parameters a' and b'. On the other hand, for tasks such as summarization, the output text is desired to contain all key information from the input Z. We consider the following loss for the summarization task f_{sum}:

$$l(f_{sum}(Z), X). \tag{3}$$

In the above formulation, the output $f_{sum}(Z)$ contains all information about X. We denote such a task as a *"complex task"*. With the squared loss, the best solution of f_{sum} is $f_{sum}(Z) = U^\top Z V^\top$. Comparing the above two tasks, for simple tasks like classification, the model just requires a small amount of information to learn a', b' $(d + D)$, while for complex tasks such as summarization, the model needs to learn $(dk + DK)$ parameters of $U' = U^\top, V' = V^\top$. Further, the model which learns more information from the data tends to memorize more. The expression in Equation 3 makes model inversion attack possible. As the model learns U', V', the attacker can conduct an attack via $Z = U'^T X V'^T$, which means one can use $f(Z)$ to recover the input data Z.

Second, the sparsity of the learned matrix (U,V) or vectors (a',b') may also vary, indicating different amounts of information needed and leading to different complexity of the task. For example, in sentiment classification, what the network actually learns depends on the sparsity of b'. If b' is sparse, it means that we can simply pick several words from the sequence and determine the class.

6.2 Attention Distributions

While the memorization disparity is possibly related to the task-specific information, the attention distribution in the transformer may also capture the contribution of each token's information to completing the task. In this section, we study whether the attention scores can be viewed as an indicator of the memorization ability of the task.

For the fine-tuned models in Table 2, we generate their attention score heatmaps[7]. Figure 4 shows the distribution of the attention scores of the last decoder-encoder attention block in each model. Different layers of decoding-encoding attention scores are also visualized in Appendix K.4[8]. Our focus on encoder-decoder attention layers is their ability to capture the information across input features for each output. We also theoretically discuss the correlation between attention score maps and task information needs in Appendix A.

In Figure 4, the horizontal axis represents input tokens, and the vertical axis indicates output tokens. The brightness represents the averaged multi-head decoder-encoder attention scores between input-output token pairs. Each horizontal line shows the attention score distribution of an output token across input tokens. We visualize the attention heatmaps of a single data sample and the average attention heatmaps of 10 random samples, padded to the longest length of the batch and truncated to a maximum of 512 tokens. More attention maps of different samples are visualized in Appendix K.3.

The heatmaps shown in Figure 4 reveal clear differences in attention patterns among tasks. For high-memorization tasks (i.e., summarization and dialog), attention scores are more evenly distributed across input tokens. In contrast, for low-memorization tasks (e.g., sentiment classification and extractive QA), the attention is concentrated on a few positions while almost zero for other positions. The observed patterns suggest that the information needed to successfully complete each task varies. The attention score distribution for summarization and dialog implies that models must extract every detail from the input, increasing the possibility of memorization. Concentrated scores for sentiment classification and extractive QA indicate that only key information is required, reducing the tendency to memorize the fine-tuning data.

In Appendix K.2, we present the attention scores of the T5-base model. Our findings also align with the fine-tuned model, and we observe a pattern of high memorization-intensive attention for complex tasks and concentrated attention for simpler ones. This aligns with our intuition that the attention pattern is a fundamental characteristic of the task. To utilize the attention score as a tool to predict memorization, prior to fine-tuning a model for a specific task, developers can assess attention patterns. The assessment can be done using the pre-trained model. It helps predict the memorization during fine-tuning.

[7]We present the heatmap of the translation task in AppendixK.1.

[8]Across various layers, consistent patterns emerge in the encoder-decoder attention mechanism. Owing to this uniformity, we primarily report on the final layer in Figure 4, which is closest to the output.

7 Conclusions

In this paper, we conduct extensive experiments to investigate the memorization behavior of fine-tuned LMs among various tasks. Utilizing an automatic detection pipeline, we are able to evaluate the memorization in numerous tasks and datasets. In addition, we provide understandings of the memorization disparity among tasks based on a sparse coding theory. Our analysis reveals a strong correlation between attention scores and memorization.

8 Limitations

Our study primarily utilizes white-box models from the T5 family for evaluation, such as T5-small, T5-base, T5-large, and T5-XL. Black-box models like ChatGPT were not included due to our inability to fine-tune them directly. Additionally, there is scope to expand the variety of datasets and tasks in future research.

Theoretically, we employ sparse coding theory to articulate our hypothesis that the observed memorization differences across tasks stem from their varying informational requirements ("True features" in sparse coding theory). We draw on recent theoretical developments(Zhang et al., 2023; Deora et al., 2023; Wu et al., 2023) that apply sparse coding in NLP and leverage these theories to support our reasoning. Nevertheless, fully extending this theory to precisely account for the complexities of non-linear or large-scale models remains an unresolved challenge in the theoretical community.

While our study addresses the memorization behavior of models fine-tuned on single tasks, the investigation into models fine-tuned on multiple tasks is still unexplored and presents an opportunity for future research.

9 Ethics Statement

The phenomenon of fine-tuned memorization in language models has notable social implications. Firstly, it raises privacy concerns, especially in tasks where sensitive information, like personal dialogue, is involved. The ability of these models to retain and potentially disclose private data necessitates corresponding data protection measures. Secondly, from a utility perspective, while memorization enhances performance in certain tasks, it also underscores the need for balancing accuracy with responsible data handling in AI systems.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2015. A latent variable model approach to pmi-based word embeddings. *arXiv preprint arXiv:1502.03520*.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raf. 2023. Emergent and predictable memorization in large language models. *arXiv preprint arXiv:2304.11158*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Puneesh Deora, Rouzbeh Ghaderi, Hossein Taheri, and Christos Thrampoulidis. 2023. On the optimization and generalization of multi-head attention. *arXiv preprint arXiv:2310.12680*.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2019. Multi-news: a large-scale multi-document summarization dataset and abstractive hierarchical model.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. 2022. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR.

Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. 2023. Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023*, pages 3637–3647.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick. 2022. Memorization in nlp fine-tuning methods. *arXiv preprint arXiv:2205.12506*.

Rishabh Misra, Mengting Wan, and Julian McAuley. 2018a. Decomposing fit semantics for product size recommendation in metric spaces. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 422–426.

Rishabh Misra, Mengting Wan, and Julian McAuley. 2018b. Decomposing fit semantics for product size recommendation in metric spaces. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 422–426, New York, NY, USA. Association for Computing Machinery.

Bruno A Olshausen and David J Field. 1997. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325.

Bruno A Olshausen and David J Field. 2004. Sparse coding of sensory inputs. *Current opinion in neurobiology*, 14(4):481–487.

R OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

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

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250.

Jie Ren, Han Xu, Pengfei He, Yingqian Cui, Shenglai Zeng, Jiankun Zhang, Hongzhi Wen, Jiayuan Ding, Hui Liu, Yi Chang, et al. 2024. Copyright protection in generative ai: A technical perspective. *arXiv preprint arXiv:2402.02333*.

Miguel A Sanchez-Perez, Alexander Gelbukh, and Grigori Sidorov. 2015. Adaptive algorithm for plagiarism detection: The best-performing approach at pan 2014 text alignment competition. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 402–413. Springer.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal

Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jingfeng Wu, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter L Bartlett. 2023. How many pretraining tasks are needed for in-context learning of linear regression? *arXiv preprint arXiv:2310.08391*.

Li Yunxiang, Li Zihan, Zhang Kai, Dan Ruilong, and Zhang You. 2023. Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge. *arXiv preprint arXiv:2303.14070*.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. 2021. Counterfactual memorization in neural language models. *arXiv preprint arXiv:2112.12938*.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. 2023. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*.

A Theoretical Analyze

To understand why attention scores can be used as an indicator for memorization, we provide some theoretical intuition on the relation between attention scores and information needed for the task. We first use the classification task to explain the relation between attention scores and information density. Then we extend the intuition to discuss complex tasks.

Attention score and memorization in classification We still consider the sparse coding model mentioned in Eq.1, and continue to use the notations of Section 6.1. We use the classification task for simplicity. As mentioned, for classification tasks, the best solution of Eq.2 is $f_{cls}(Z) = (a')^\top Z b'$, where $a^\top U^\top := a'$ and $V^\top b := b'$. In classification, whether a task is more complex or not depends on the sparsity of b', and we justify that the **sparsity of b' directly affects the attention score pattern.** We then mathematically define the neural network architecture. To ease the derivation, we consider

$$f(Z) = W^V Z \cdot \text{softmax}\left((W^K Z)^\top (W^Q Z)\right), \tag{4}$$

with W^V, W^K, W^Q all in $\mathbb{R}^{d\times d}$. The softmax operation is conducted column-wise. Since the output $f(Z)$ is a matrix rather than a scalar, for the classification task, we further multiply two vectors on the two sides of $f(Z)$ to get output scalar y', i.e., $v_1^\top f(Z) v_2 \in \mathbb{R}$, and v_1 and v_2 can be either trainable or arbitrary. As mentioned in Section 6.1, the target of the classification task can be represented as $y = a^\top X b$. The loss term of Eq.2 can then be written as:

$$l(v_1^\top f(Z) v_2, a^\top X b). \tag{5}$$

Aligning the neural network output $v_1^\top f(Z) v_2$ with $a^\top X b$, it is easy to see that to better reduce the loss value, we need softmax $\left((W^K Z)^\top (W^Q Z)\right) v_2 \in \mathbb{R}^D$ better aligned with b'. As a result, when measuring the effect of the input Z on $v_1^\top f(Z) v_2 \in \mathbb{R}$, e.g., Figure 4e, the weighted pattern softmax $\left((W^K Z)^\top (W^Q Z)\right) v_2$ has a similar sparsity as b'. Recall that b' is the task-specific vector the model needs to learn, thus the analysis above suggests that **the attention score[9] has a similar sparsity pattern as the sparsity of the information the model needs to learn.**

Complex tasks For more complex tasks, the simplistic single-layer single-head attention analysis as the above is not enough to handle it, and we need to use a larger architecture. Intuitively, with more features to learn in the task, the architecture will be more likely to memorize each feature comprehensively. We identify two key drivers of this behavior. First, each output token relies on information distributed across multiple input tokens. As shown in Figure 4a, each row has multiple high attention scores across different input tokens. Second, each output token often exhibits selectivity for a different subset of input tokens, leading to divergence in attention distributions across rows in Figure 4a. To conclude, these two factors may result in dense heatmap patterns compared to the concentrated heatmaps of simpler tasks.

[9]Note that the attention matrix softmax $\left((W^K Z)^\top (W^Q Z)\right)$ itself is $\mathbb{R}^{D\times D}$ and is not aggregated for the output value, and softmax $\left((W^K Z)^\top (W^Q Z)\right) v_2$ is the final aggregated attention.

B Details of Evaluation Pipeline

B.1 Evaluation Process

In this section, we provide a detailed overview of the evaluation tools and methodologies employed. The evaluation framework comprises three distinct processes: prompting, searching, and detection. The prompting method has been extensively utilized in prior research on pre-trained memorization(Carlini et al., 2022), and the search and detection methods, based on Elasticsearch and PAN-2014 detection tools, were previously adopted by Lee et al. (2023).

Prompting. In the evaluation phase, the input data x_i is segmented into two parts: a prefix p_i and a suffix s_i. We input the prefixes $\{p_i\}_{i=1}^n$ into the model without any task-specific instructions to obtain $f(p_i)$. For our experiments, we select $n = 10,000$ samples from each dataset. The standard prefix length k is set to 50 tokens. However, for tasks with input sentences shorter than 50 tokens, such as translation tasks, a reduced prefix length of 15 tokens is used. The testing procedure is consistent across both base and fine-tuned models. This involves inputting the prefix of the fine-tuning data into the models and comparing the suffixes. Notably, for the base models, the prefix of the fine-tuning data is still used in the test, regardless of whether the model has been fine-tuned on that specific dataset or not. This approach helps determine if the model retains the data, suggesting

its presence in the pre-training set.

Searching. In the search phase of our experiment, we employ Elasticsearch, a distributed, RESTful search and analytics engine based on the open-source Lucene library. Elasticsearch leverages the Okapi-BM25 algorithm, a widely-used bag-of-words ranking function, allowing for efficient storage, searching, and near real-time analysis of large data volumes. We upload all suffixes $\{s_i\}_{i=1}^n$ into Elasticsearch and use the set $\{f(p_i)\}_{i=1}^n$ as our query documents. For our analysis, we set $K = 10$, indicating that only the top-10 most relevant candidates for each query are retrieved for subsequent memorization detection.

Detection. After we get suspicious sentence pairs $f(p_i)$ and s_j, we input them to a publicly available PAN2014 plagiarism detection tool D to see if $D(f(p_i), s_j) =$ True. In general, the detection tool will detect the presence of plagiarised word piece pairs (d_i, d_j) a, where d_i and d_j are word pieces from $f(p_i)$ and s_j, respectively and then compare (d_i, d_j) to identify the category of memorization. Here we set the minimal match threshold as at least 50 characters(approximately 15 tokens).

The detailed process includes (1) preprocessing text; (2) identifying obfuscation types; (3) seeding to find candidate pairs via sentence similarity; (4) extension by clustering similar fragments; and (5) filtering out overlaps. They transform sentences into TF-IDF vectors and calculate similarity using dice and cosine measures, with adaptive parameters selected by testing on the obfuscation corpus. Here we set the minimal match threshold as at least 50 characters(approximately 20 tokens). We also utilize additional validation steps after retrieving paraphrased text segments as (Sanchez-Perez et al., 2015). The post-processing involves chunking segments into sentences using NLTK's tokenizer, then applying a RoBERTa-based paraphrase identification model and Named Entity Recognition (NER) on the sentences. Specifically, we check sentence pairs - if any pair has a paraphrase detection probability score between 0.5 and 0.99, we accept it as high-confidence paraphrasing, otherwise, we identify it as low-confidence paraphrasing.

B.2 Memorization Types

Difference between memorization types. Here we will distinguish 3 types of memorization. First, verbatim memorization means exact copies of words or phrases without transformation. In the cases of paraphrase and idea memorization, the output is not identical to the original text but shares similar meanings. While paraphrase plagiarism focuses on sentence-to-sentence transformation, idea plagiarism involves summarizing the key points of a larger text segment into a more condensed form (or expanding it). In Table 3, we give a simple example to differentiate the difference between 3 types of memorization conceptually.

In practice of the PAN2014-detection, It starts by identifying closely matched short document fragments (referred to as 'seeds') and then expands these seeds into longer text segments. This is achieved by clustering these fragments based on their separation and employing a 'maxgap' threshold parameter to form coherent clusters.They experimentally find out the most suitable threshold for different plagiarism datasets so that those parameters could be used for the detection of a specific type of memorization. In other words, each memorization case will be **counted only once** and there will not be overlapping across different categories.

Distinguishing idea memorization from summarization. It's important to differentiate idea memorization—*condensing key points of a larger text segment*—from summarization tasks. Note that in our approach, only the prefix (initial tokens) of a text is input to the model. Summarization means the model summarizing this prefix without revealing the remaining suffix of the text. In contrast, idea memorization involves the model generating information about the suffix. In our experiments, we assess similarity by comparing the generated text $f(p)$ with the suffix s, rather than with the prefix p or the entire text x. In table 4, we use a simple example to illustrate the difference.

C Disparate Memorization on GPT-Neo

We also fine-tuned decoder-only GPT-Neo-125m models and also observed similar findings with T5-base, which suggests our findings are generalizable. The results are reported in Table 9. We can clearly observe that the memorization increase for summarization and dialog is much more significant than QA and sentiment classification.

D Memorization Ratio on Other Tasks and Datasets

To make a general conclusion, we would also like to provide memorization behavior on additional datasets and tasks, including summarization

Table 3: Difference between 3 Memorization Types

Examples of 3 Memorization Types
Verbatim:
Text A: My name is Jack
Text B: My name is Jack
Paraphrase:
Text A: My name is Jack
Text B: Jack is my name
Idea plagiarism:
Text A: A boy tell me in the class that his name is Jack
Text B: A boy is Jack

Table 4: Difference between Idea Memorization and Summarization

Idea Memorization vs Summarization
Training data: I am not comfortable from the beginning of the month, I am 20 years old, height 51, height 51, weight 40kg. I have been pregnant for 6 months and I can not stop vomiting.
Input: I am not comfortable from the begining of the month,I am 20 years
Output: 20 years old, height 5 1 & weight 40kg(Memorization)
Output: 20 years old woman feels bad for a month.(Summarization)
Note: We only compare output with *"height 51, height 51, weight 40kg. I have been pregnant for 6 months and I can not stop vomiting."* to identify memorization.

(CNN_Daily Mail), abstractive question answering (DuoRC Self_RC), extractive question answering (Adversarial_QA), multiple choice (BoolQ), and topic classification (AG_News).

Similar to the experiments in the main paper, we observe that tasks with dense attention maps (e.g., summarization) exhibit high memorization, while tasks with sparse attention (e.g., extractive QA, abstractive QA, multiple-choice QA, topic classification) display low memorization. These results validate the generalizability of our observations.

E Memorization of Multi-task Fine-tuned Models

We conducted a preliminary study on the memorization of multi-task fine-tuned models. Specifically, we compared the memorization ratio on the Multi_News dataset between a T5-base model fine-tuned solely on that dataset and Flan-T5, a multi-task fine-tuned model that includes the Multi_News dataset in its training set. Our results on Table 11 reveal that multi-task fine-tuned models exhibit significantly lower memorization compared to single-task fine-tuned models. These findings suggest that multi-task fine-tuning could be a potential mitigation strategy against memorization in language models.

F Ablation Studies

F.1 Training Epochs

Here, we present the memorization rates observed in checkpoints of our fine-tuned models across different epochs, as shown in Table 5.

In our practice, we find that for low-memorization tasks like sentiment classification, no matter whether the model is well-trained, the memorization ratio remains low. However, for high-memorization tasks like dialog, if the model is not well-trained, the memorization will be low. So in our experiment, to make sure that our fine-tuned model is well-trained, we let the finetuned model have comparable performance with Flan-T5 as Flan-T5 is a well-trained model and has good performance on various tasks.

F.2 Sampling Methods

We conduct ablation studies on different decoding methods in Table 6. From the results, we can find that:

- For high-memory tasks such as summarization and Dialog, sampling can reduce the memorization Rate and change the category distribution of memory samples.
- For low-memory tasks such as emotion classi-

Table 5: Memorization of fine-tuned_T5 with various epochs.

Task	Dataset	Epochs	Total Mem Rate
Dialog	HealthCareMagic	1	1.60%
		3	3.30%
		5	6.22%
		10	8.27%
Sentiment	IMDB	1	0.78%
		3	0.79%
		5	0.80%
		10	0.79%
Summarization	Multi_news	1	14.12%
		3	14.32%
		5	22.33%
		10	22.32%
QA	Squad_v2	1	0.08%
		3	0.12%
		5	0.10%
		10	0.15%

Table 6: Memorization of fine-tuned_T5 with various sampling methods.

Task	Dataset	Decoding	Total Mem Rate	Verbatim	Idea	Paraphrase ($P < 0.5$)	Paraphrase ($P > 0.5$)
Dialog	HealthCareMagic	Top-K	5.76%	0.05%	0.38%	0.90%	4.43%
		Top-p	7.26%	0.06%	0.48%	1.35%	5.37%
		Temp	3.72%	0.02%	0.18%	0.58%	2.94%
		Greedy	8.27%	0.02%	1.41%	1.75%	5.09%
Sentiment	IMDB	Top-K	1.02%	0.01%	0.13%	0.18%	0.70%
		Top-p	1.08%	0.01%	0.12%	0.22%	0.73%
		Temp	0.89%	0.01%	0.07%	0.19%	0.62%
		Greedy	0.80%	0.04%	0.30%	0.17%	0.29%
Summarization	Multi_news	Top-K	10.80%	2.54%	0.34%	1.94%	5.98%
		Top-p	13.57%	4.07%	0.54%	2.26%	6.70%
		Temp	5.82%	1.28%	0.23%	0.83%	3.48%
		Greedy	22.33%	4.23%	0.65%	6.23%	11.22%
QA	Squad_v2	Top-K	0.15%	0.04%	0.00%	0.05%	0.06%
		Top-p	0.15%	0.04%	0.00%	0.05%	0.06%
		Temp	0.15%	0.04%	0.00%	0.05%	0.06%
		Gready	0.15%	0.04%	0.00%	0.05%	0.06%

Table 7: Memorization of fine-tuned T5 with varying prefix lengths.

Task	Dataset	Prefix length	Total Mem Rate	Verbatim	Idea	Paraphrase ($p > 0.5$)	Paraphrase ($p < 0.5$)
Summarization	Multi_news	10	12.25%	1.74%	2.85%	0.88%	6.78%
		30	20.68%	7.07%	1.41%	3.05%	9.15%
		50	22.33%	4.23%	0.65%	6.23%	11.22%
		100	29.66%	10.61%	0.79%	4.27%	13.99%
Dialog	HealthCareMagic	10	6.28%	0.03%	1.94%	0.85%	3.46%
		30	7.76%	0.04%	1.28%	1.72%	4.72%
		50	8.27%	0.02%	1.41%	1.75%	5.09%
Sentiment	IMDB	10	1.37%	0.00%	1.12%	0.06%	0.19%
		30	1.18%	0.01%	0.51%	0.15%	0.51%
		50	0.80%	0.04%	0.30%	0.17%	0.29%
		100	1.39%	0.05%	0.23%	0.33%	0.78%
QA	Squad_v2	10	0.18%	0.05%	0.00%	0.06%	0.07%
		30	0.14%	0.04%	0.00%	0.04%	0.06%
		50	0.15%	0.04%	0.00%	0.05%	0.06%
		100	0.13%	0.03%	0.01%	0.04%	0.05%

fication, sampling does not significantly affect the memorization results.

- Irrespective of the decoding methodology employed, **a pronounced disparity in memorization across different tasks persists.** This suggests an inherent task-specific propensity towards memorization that is not substantially mitigated by variations in sampling techniques.

F.3 Prefix Lengths

Here we change different prefix lengths of inputs and report the results in table 7. We include 2 high-memorization tasks(summarization and dialog) and 1 low-memorization task (sentiment classification). From the results we can observe that:

- **The length of prefix tokens can affect memorization.** The length of prefix tokens does indeed impact memorization. Specifically, for summarization and Dialog tasks, the memorization Rate generally increases with the length of the prefix. This finding aligns with previous research on pre-trained memorization. However, for sentiment classification, changing the prefix does not result in significant changes, and increasing the prefix length does not necessarily lead to an increase in the memorization Rate.

- **The task disparity still exists when using different prefixes.** Furthermore, it is worth noting that despite the influence of different prefixes on memorization, there still exists a noticeable disparity in memorization across tasks. Therefore, our conclusion remains even using different prefixes.

G Statistical Significance Testing

In this section, the results displayed in Tables 12 to 15 are derived from the data in Tables 1, 2, 6, 7, and Figure 2. Specifically, we conducted 1000 bootstrap experiments based on these sources to calculate confidence intervals at the 5% and 95% levels for the results presented in the aforementioned tables.

H Dataset and Model used

Datasets In our study, we utilized various datasets for preliminary experiments and model fine-tuning. For the summarization task, we used CNN/Daily Mail in the preliminary phase, a dataset comprising 287k training rows and 10k evaluation rows. For fine-tuning, we employed the **Multi-News** dataset (Fabbri et al., 2019), which includes news articles and summaries, using a 45k training set and a 5.62k test set. The **IMDB** dataset (Maas et al., 2011) was used for binary sentiment classification, consisting of 25k training and 25k test movie reviews. In the dialog task, we use **HealthcareMagic** dataset, comprising 112k training rows

and a 12k test set. For extractive QA, we utilized the **SQuAD** v2 dataset (Rajpurkar et al., 2016), featuring questions based on Wikipedia articles, with 130k training and 11.9k test rows. The translation task involved a preliminary study using **WMT19** and fine-tuning on an English-to-German subset of **WMT16**, with 450.87k training and 3k test rows. Finally, for the controlling experiment, we used the **RentTheRunway** dataset (Misra et al., 2018b), containing clothing review data, with 111k training and 12k test rows. This dataset was used for fine-tuning both the summarization model and the binary sentiment classification task.

Models In the preliminary study, we consider Bart-Large from Bart family, T5-base and T5-large from T5 family, FSMT (FairSeq MachineTranslation), and BioGPT. For our self-fine-tuned models, we select T5-base architecture from the T5 family for all experiments.

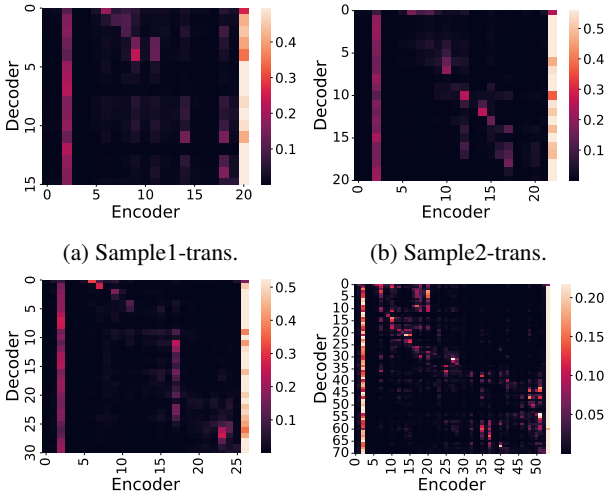
Fine-tuned Methods In the fine-tuning process, we consider all the tasks as generation tasks and use the format of instruction tuning. Here we provide fine-tuned templates of different tasks in Table 8

I Performance of Self-fine-tuned Models

We finetune the T5-base model to achieve better or comparable performance with the Google fine-tuned public model FLAN-T5. In Table 16, we show the performance of the summarization task. Our fine-tuned model achieves a similar rouge score with FLAN-T5. In Table 17, We show that the accuracy of our model is better than FLAN-T5 regarding binary sentiment classification. For Dialog task, Our model performance much better than FLAN-T5 as shown in Table 18. For the Extractive question-answering task, we fine-tune the model in a sequence-to-sequence learning form while we evaluate the exact match of the answer term. Results are shown in Table 19. For the RentTheRunway fine-tuning experiment, we present the results in Table 20 and Table 21.

J Memorized Examples

We present memorization examples of verbatim, paraphrase and idea plagiarism of different models in Table 22.



(a) Sample1-trans. (b) Sample2-trans.

(c) Sample3-trans. (d) Sample4-trans.

Figure 5: Decoder-encoder attention heatmaps on translation.

K Attention Maps

K.1 Attention Maps of Translation Tasks

In Figure 5, we present the attention maps for translation tasks. The visualized examples reveal that attention scores tend to focus on a select few input features corresponding to each output token. This concentration of attention is likely due to the nature of translation tasks, where the model typically does not require full detail attention but rather focuses on key tokens for accurate translation. As a result, attention is directed predominantly towards specific features, leading to reduced overall memorization. These observations are consistent with our findings from other tasks.

K.2 Attention Maps of T5-base when doing different tasks

To validate that attention patterns are more intrinsic properties of the tasks themself, we visualize the attention maps of the T5-base model(without fine-tuning) when doing different tasks in Figure 6. Specifically, we use the same instruction and input-output pairs of fine-tuning data as Section 5.1, but just change the model from finetuned-T5 to T5-base. From the Figure we can see that the disparity still exists across different tasks. And for each task, the attention patterns are similar to that of Fine-tuned T5. It further validates that the information needed to complete certain tasks is the intrinsic property of the task.

K.3 Attention Maps of different samples

In this section, we extend our visualization of attention maps across a broader range of samples and tasks, from Figure 7 to Figure 10. It is evident that memorization patterns differ significantly among tasks. Tasks with higher memorization requirements, such as summarization, display densely distributed attention scores, while those with lower memorization needs, like Extractive QA, exhibit more focused attention distributions.

K.4 Attention Maps of Different encoder-decoder layers

Here we visualize the attention maps of different encoder-decoder layers in Figure 11 to Figure 14. We can clearly observed consistent patterns across various layers of the encoder-decoder attention mechanism, with high memorization tasks showing dense attention and low memorization tasks focusing attention on fewer positions.

Table 8: Examples of training data from different tasks

Summarization

Prompt: Please summarize the following paragraph:

Input:

...A fresh update on the U.S. employment situation for January hits the wires at 8:30 a.m. New York time offering one of the most important snapshots on how the economy fared during the previous month. Expectations are for 203,000 new jobs to be created, according to economists polled by Dow Jones Newswires, compared to 227,000 jobs added in February. The unemployment rate is expected to hold steady at 8.3%. ...

Output:

...The unemployment rate dropped to 8.2% last month, but the economy only added 120,000 jobs, when 203,000 new jobs had been predicted, according to today's jobs report.

Training Format:

Training input = **Prompt** + **Input**, Training label = **Output**

Sentiment classification

Prompt:

Please classify the sentiment of the following paragraph:

Input:

"Foxes" is a serious look at the consequences of growing up too fast in the 1980s. And unlike the teen sex comedies that overshadowed it (Porky's, Fast Times at Ridgement High), the movie holds up well against time...

Output:

Positive

Training Format:

Training input = **Prompt** + **Input**, Training label = **Output**

Dialog

Instruction:

If you are a doctor, please answer the medical questions based on the patient's description.

Input:

I woke up this morning feeling the whole room is spinning when i was sitting down. I went to the bathroom walking unsteadily, as i tried to focus i feel nauseous. I try to vomit but it wont come out.. After taking panadol and sleep for few hours, i still feel the same..

Output:

Hi, Thank you for posting your query. The most likely cause for your symptoms is benign paroxysmal positional vertigo (BPPV), a type of peripheral vertigo. In this condition, the most common symptom is dizziness or giddiness, which is made worse with movements. ...

Training Format:

Training input = **Instruction** + **Input**, Training label = **Output**

Question and answering

Question:

Who was the Norse leader?

Input:

... They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. ...

Output:

Rollo

Training Format:

Training input = **Question** + **Input**, Training label = **Output**

Table 9: Memorization rate of gpt-neo-125m fine-tuned on various tasks

Task	Dataset	Model	Total Mem Rate
Summarization	Multi-news	GPT-Neo	25.2%
		GPT-Neo-ft	44.3%
		Difference	↑19.1%
Dialog	chatdoctor	GPT-Neo	2.5%
		GPT-Neo-ft	7.8%
		Difference	↑5.3%
Sentiment Classification	imdb	GPT-Neo	3.9%
		GPT-Neo-ft	4.2%
		Difference	↑0.3%
Reading Comprehension	Squad_v2	GPT-Neo	0.02%
		GPT-Neo-ft	0.04%
		Difference	↑0.02%
Translation	wmt	GPT-Neo	0.00%
		GPT-Neo-ft	0.00%
		Difference	-

Table 10: Memorization rate of Other Tasks and Datasets

Source Model	Task	Dataset	Total Mem rate
T5-base	Summarization	CNN_Daily	5.82%
T5-base	Extractive QA	Adversarial_QA	0.00%
T5-base	Abstractive QA	Duorc_SelfRC	0.02%
T5-base	Multiple Choice	BoolQ	0.13%
T5-base	Topic Classification	AG_News	0.03%

Table 11: Memorization rate of multi-task fine-tuned Flan-T5

Task	Model	Dataset	Total Mem rate
Summarization	Flan-T5	Multi_news	4.96%
Summarization	T5-finetuned	Multi_news	22.32%

Table 12: Statistical significance test of open-sourced LLMs fine-tuned on various tasks

Task	Dataset	Source Model	Total Mem Rate (CI)
Summarization	CNN/Daily Mail	Bart_Large	[20.30%, 21.10%]
Medical Dialog	ChatDoctor	BioGPT	[18.86%, 20.36%]
Extractive QA	SQuAD_v2	T5_large	[0.05%, 0.17%]
Abstractive QA	Race	T5_large	[0.19%, 0.41%]
Translation	WMT_19	FSMT	[0%, 0%]
Sentiment Classification	IMDB	T5-base	[0%, 0%]

Table 13: Statistical significance test of T5-base fine-tuned on various tasks

Task	Dataset	Model	Total Mem Rate (CI)
Summarization	Multi-news	T5-base	[13.28%, 14.68%]
		T5-finetuned	[21.52%, 23.14%]
		Difference	[7.84%, 8.86%]
Dialog	chatdoctor	T5-base	[1.36%, 1.84%]
		T5-finetuned	[7.75%, 8.81%]
		Difference	[6.15%, 7.19%]
Sentiment Classification	imdb	T5-base	[0.61%, 0.95%]
		T5-finetuned	[0.62%, 0.98%]
		Difference	[0%, 0.04%]
Reading Comprehension	Squad_v2	T5-base	[0.07%, 0.09%]
		T5-finetuned	[0.07%, 0.23%]
		Difference	[0.01%, 0.19%]
Translation	wmt	T5-base	[0%, 0%]
		T5-finetuned	[0%, 0%]
		Difference	[0%, 0%]

Table 14: Statistical significant test of fine-tuned T5 with various sampling methods.

Task	Dataset	Decoding	Total Mem Rate (CI)
Summarization	Multi_news	Top-K	[10.17%, 11.43%]
		Top-p	[12.89%, 14.25%]
		Temp	[5.37%, 6.27%]
		Greedy	[21.48%, 23.18%]
Dialog	HealthCareMagic	Top-K	[5.32%, 6.23%]
		Top-p	[6.78%, 7.77%]
		Temp	[3.34%, 4.08%]
		Greedy	[7.75%, 8.79%]
Sentiment	IMDB	Top-K	[0.83%, 1.23%]
		Top-p	[0.87%, 1.29%]
		Temp	[0.71%, 1.08%]
		Greedy	[0.63%, 1.00%]
QA	Squad_v2	Top-K	[0.08%, 0.23%]
		Top-p	[0.07%, 0.23%]
		Temp	[0.08%, 0.23%]
		Greedy	[0.08%, 0.23%]

Table 15: Statistical significance test of fine-tuned T5 with varying prefix lengths.

Task	Dataset	Prefix length	Total Mem Rate (CI)
Summarization	Multi_news	10	[11.94%, 12.56%]
		30	[20.27%, 21.09%]
		50	[21.99%, 22.67%]
		100	[29.22%, 30.10%]
Dialog	HealthCareMagic	10	[5.79%, 6.77%]
		30	[7.23%, 8.29%]
		50	[7.76%, 8.78%]
Sentiment	IMDB	10	[1.13%, 1.61%]
		30	[0.96%, 1.40%]
		50	[0.63%, 0.97%]
		100	[1.15%, 1.63%]
QA	Squad_v2	10	[0.09%, 0.27%]
		30	[0.06%, 0.22%]
		50	[0.07%, 0.24%]
		100	[0.06%, 0.21%]

Table 16: Summarization

Dataset	Model	Rouge1	Rouge2	RougeL	RougeLSum
multi_news	FLAN-T5-small	0.264	0.092	0.168	0.168
multi_news	Our fine-tuned T5-small	0.308	0.088	0.187	0.187
multi_news	FLAN-T5-base	0.291	0.098	0.237	0.237
multi_news	Our fine-tuned T5-base	0.298	0.103	0.201	0.201
multi_news	FLAN-T5-large	0.256	0.087	0.165	0.165
multi_news	Our fine-tuned T5-large	0.368	0.122	0.218	0.218
multi_news	FLAN-T5-3b	0.264	0.092	0.232	0.232
multi_news	Our fine-tuned T5-3b	0.387	0.136	0.168	0.168

Table 17: Sentiment classification

Dataset	Model	Accuracy(%)
IMDB	FLAN-T5-small	94.17
IMDB	Our fine-tuned T5-small	95.30
IMDB	FLAN-T5-base	93.56
IMDB	Our fine-tuned T5-base	94.64
IMDB	FLAN-T5-large	94.50
IMDB	Our fine-tuned T5-large	95.30
IMDB	FLAN-T5-3b	97.10
IMDB	Our fine-tuned T5-3b	95.20

Table 18: Dialog

Dataset	Model	Rouge1	Rouge2	RougeL	RougeLSum
HealthCareMagic	FLAN-T5-small	0.041	0.004	0.031	0.031
HealthCareMagic	Our fine-tuned T5-small	0.131	0.063	0.154	0.154
HealthCareMagic	FLAN-T5-base	0.055	0.006	0.039	0.039
HealthCareMagic	Our fine-tuned T5-base	0.298	0.103	0.201	0.201
HealthCareMagic	FLAN-T5-large	0.068	0.007	0.050	0.050
HealthCareMagic	Our fine-tuned T5-large	0.220	0.063	0.154	0.154
HealthCareMagic	FLAN-T5-3b	0.073	0.010	0.055	0.055
HealthCareMagic	Our fine-tuned T5-3b	0.139	0.012	0.094	0.094

Table 19: Question answering

Dataset	Model	Exact Match(%)
SQuAD v2	FLAN-T5-small	35.23
SQuAD v2	Our finetuned T5-small	49.1
SQuAD v2	FLAN-T5-base	34.30
SQuAD v2	Our finetuned T5-base	44.00
SQuAD v2	FLAN-T5-large	43.36
SQuAD v2	Our finetuned T5-large	53.20
SQuAD v2	FLAN-T5-3b	44.60
SQuAD v2	Our finetuned T5-3b	37.20

Table 20: Multi task trained with RentTheRunway

Dataset	Model	Rouge1	Rouge	RougeL	RougeSum
Summary	FLAN-T5-Base	0.1743	0.0436	0.1598	0.1598
Summary	Our finetuned T5-Base	0.1743	0.0436	0.1598	0.1598

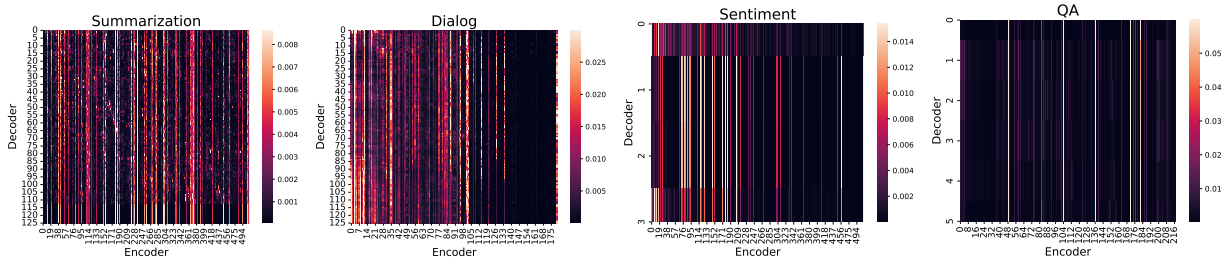
Table 21: Multi task trained with RentTheRunway

Dataset	Model	Accuracy(%)
Sentiment Classification	Flan T5-base	86.60
Sentiment Classification	Our fine-tuned T5-base	98.07

Table 22: Examples of memorization cases. Duplicated texts are highlighted with yellow marks. Personally identifiable information (PII) and other words that may lead to privacy concern in generated text are masked as red.

Type	Machine-Written Text	Training Text
Paraphrase	At least10 people and two attackers were killed in Tuesday's attack against the luxurious Corinthia Hotel in Tripoli, Libya, a spokesman for a security division of the Ministry of Interior in Tripoli said. Five foreigners – one American (***Summarization: CNN Daily, Bart Large***)	On January 27, gunmen claiming to be affiliated with ISIS attacked the Corinthia Hotel in Tripoli, Libya, which is favored by government officials and foreigners. They killed 10 people after storming into the lobby and firing guns at hotel guests. Five of the victims were foreigners, one an American.
Paraphrase	– Argentine President Cristina Fernandez de Kirchner was told to take a month off work after doctors diagnosed her with a subdural hematoma. (***Summarization: CNN Daily, Bart Large***)	...hematoma and said she needed to take a month off of work.
Paraphrase	I am a 20-year-old guy 20 years old. I have been (***harassment word***) for a long time. (***Dialog: HealthCareMagic, Finetuned T5***)	...20 years old i have been. (***harassment word***) regularly for past 5 years
Paraphrase	The Chargers responded with a 1-yard TD run by RB LaDainian Tomlinson. (***Abstractive QA: Finetuned T5***)	The Chargers would respond with RB LaDainian Tomlinson with a 4-yard TD run.
Paraphrase	– President Trump has named Mick Mulvaney to replace John Kelly, the White House chief of staff who left the White House in December." (**Summary: Multi news, Finetuned T5**)	"I am pleased to announce that Mick Mulvaney, Director of the Office of Management & Budget, will be named Acting White House Chief of Staff, replacing General John Kelly, who has served our ...
Verbatim	Rachel's son Liam in a house near Glenrothes on 22 March 2014. (***Summary: CNN Daily, Finetuned T5***)	Rachel's son Liam in a house near Glenrothes on 22 March 2014.
Verbatim	divided Wednesday during heated arguments over President Obama's health care law, but (***Summary: Multi news, Finetuned T5***)	divided Wednesday during heated arguments over President Obama's health care law, but
Verbatim	and liver cirrhosis in dec 2011 modified akt staarted because of cirrhosis i.e (***Dialog:Finetuned T5, ChatDoctor***)	and liver cirrhosis in dec 2011 modified akt staarted because of cirrhosis i.e

Type	Machine-Written Text	Training Text
Verbatim	River Martinez, 10, breaks camp at the Upper Pines Campground in Yosemite National Park, Calif., on Wednesday, July 25, 2018. (***Summary: Multi news, Finetuned T5***)	River Martinez, 10, breaks camp at the Upper Pines Campground in Yosemite National Park, Calif., on Wednesday, July 25, 2018.
Verbatim	A rare blue lobster caught by local lobsterman, Greg Ward, is on display at the Seacoast Science Center in Rye, N.H., on Tuesday, July 18, 2017. (***Summary: Xsum, Finetuned T5***)	A rare blue lobster caught by local lobsterman, Greg Ward, is on display at the Seacoast Science Center in Rye, N.H., on Tuesday, July 18, 2017.
Verbatim	Sheffield homered twice and keyed a four-run rally in the ninth inning Thursday night, sending the (***Classification: AG news, Finetuned T5***)	Sheffield homered twice and keyed a four-run rally in the ninth inning Thursday night, sending the
Idea	KUALA LUMPUR (Reuters) - Kim Jong Un's half-brotherwas carrying $100,000 in cash in his backpack at the time of his murder, the officer investigating the case told a police officer" (***Summary: Multi news, Finetuned T5***)	Wan Azirul testified that Kim was carrying $100,000 in cash in his backpack.
Idea	Alan Dawson, 64, of Urmston, was convicted of seven counts of indecent assault and one count of rape at Manchester Crown Court. (***Summary: Xsum, Finetuned T5***)	...is charged with one count of rape and one count of sexual assault.
Idea	– Trey Radel, the Florida Rep. who was arrested last month for buying cocaine, is a freshman congressman who has been a big news story for the Washington Post. (***Summary: Multi News, Finetuned T5***)	Post) \n \n Florida Rep. Trey Radel (R-Fla.) was arrested last month for buying cocaine.
Idea	Abdul Aziz believes he was standing right next to a shooter when gunmen opened fire at a parade in new orleans, injuring 19 people. "Everyone around me was right next to a shooter," Abdul Aziz said. (***Summary: CNN Daily, Finetuned T5***)	I was standing, I believe, right next to the shooter.



(a) (b) (c) (d)

Figure 6: Average decode-encoder attention heatmaps on (a) summarization, (b) dialog, (c) sentiment, and (d) QA on T5-base across 10 samples

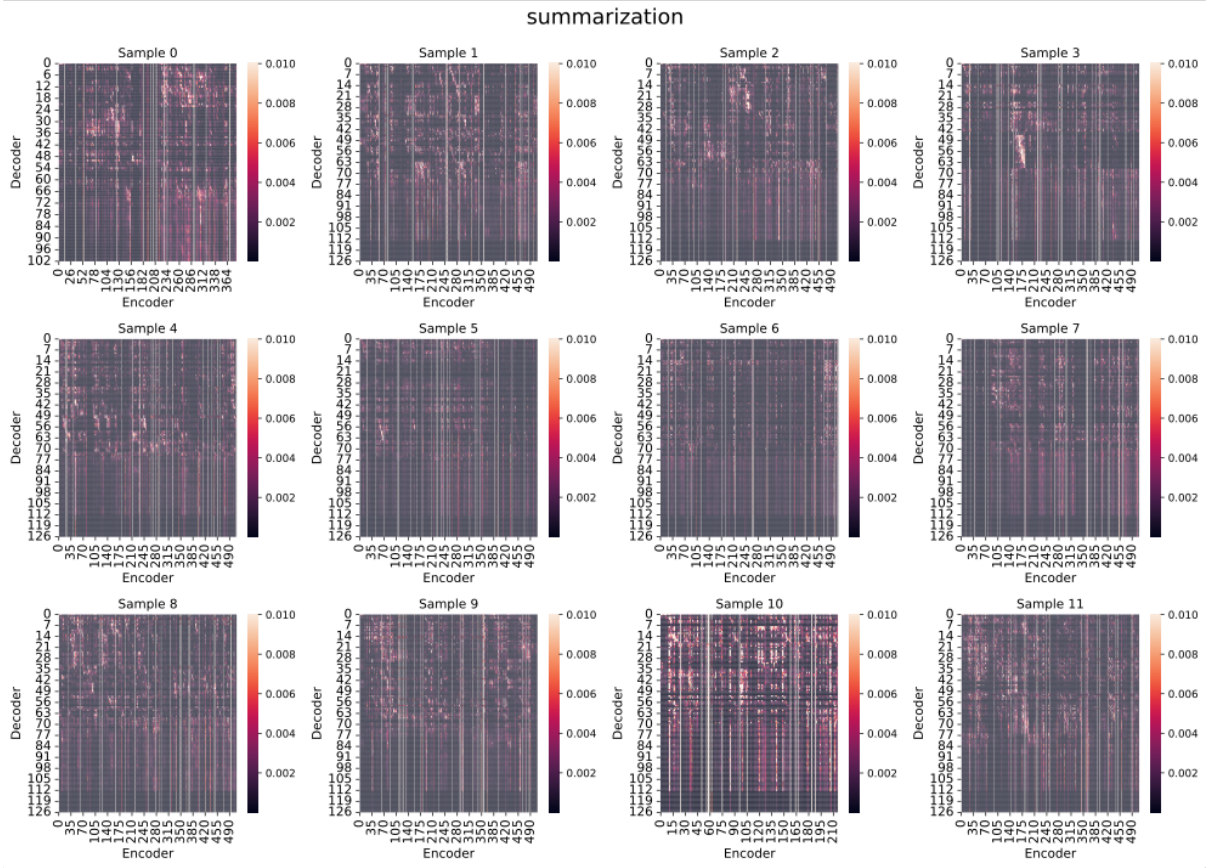


Figure 7: Attention heatmaps of different samples on summarization

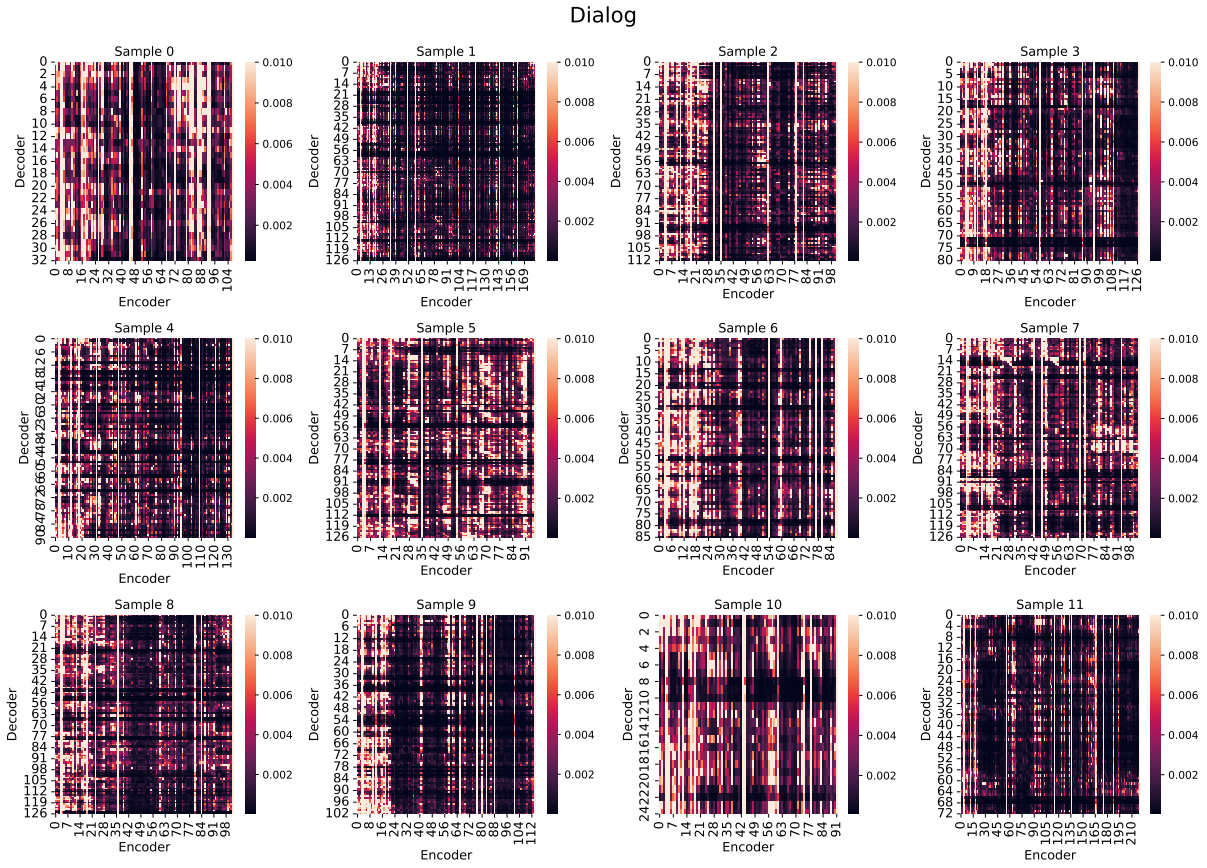


Figure 8: Attention heatmaps of different samples on dialog

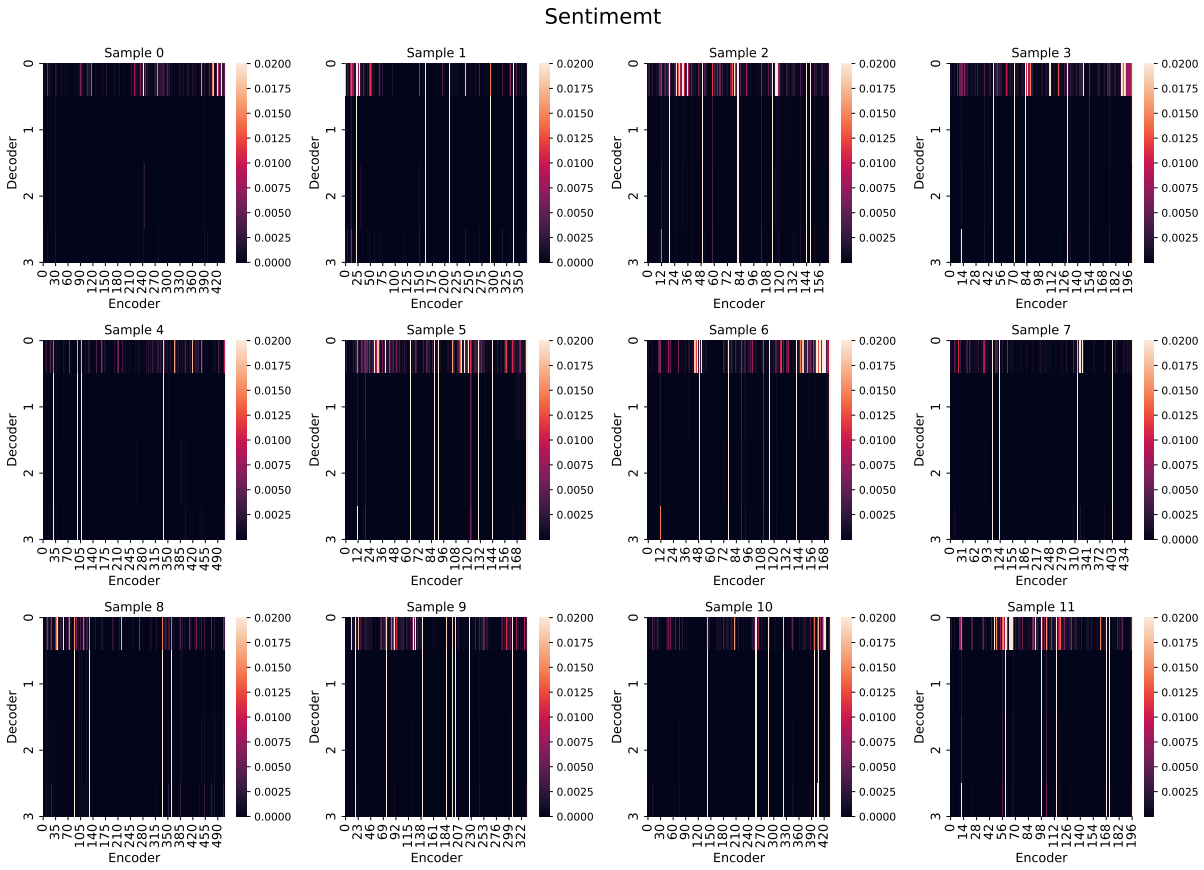


Figure 9: Attention heatmaps of different samples on sentiment classification

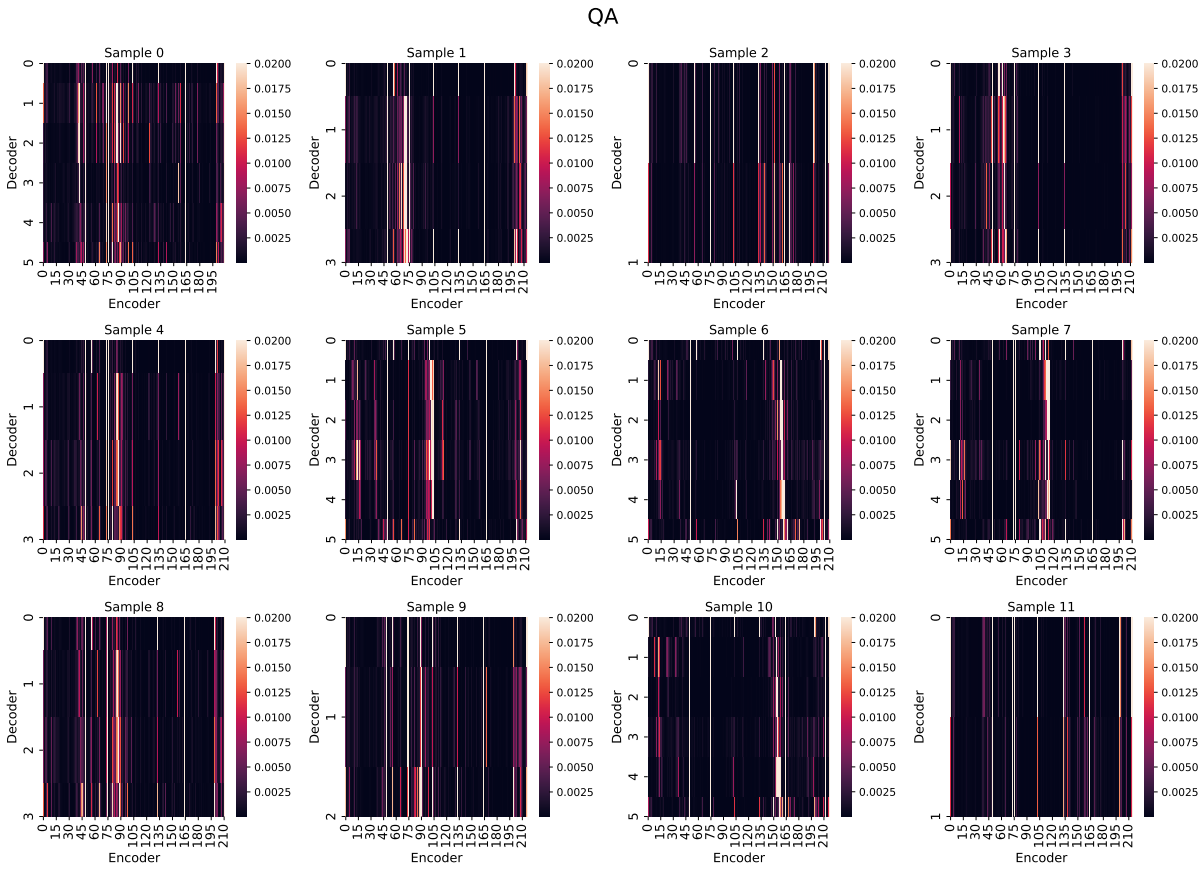


Figure 10: Attention heatmaps of different samples on QA

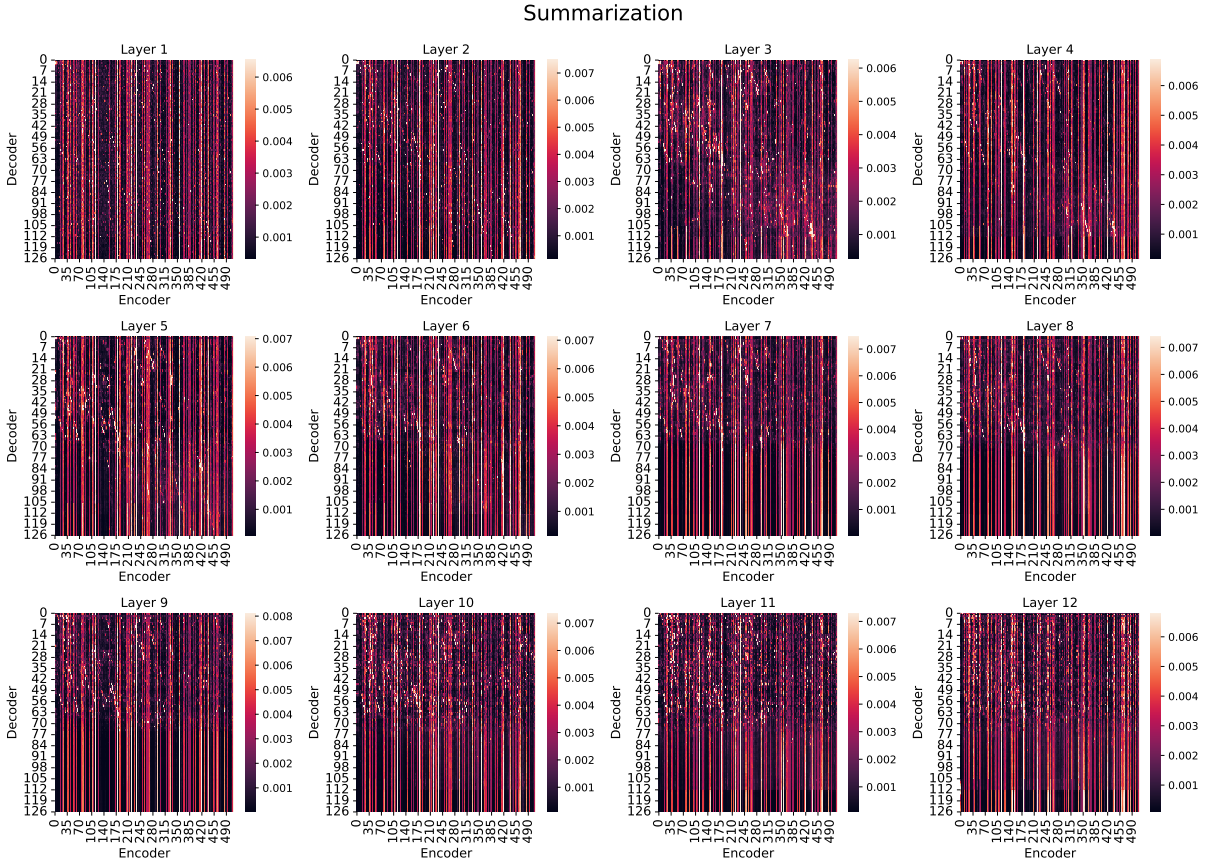


Figure 11: Average decode-encoder attention heatmaps on summarization from different layers

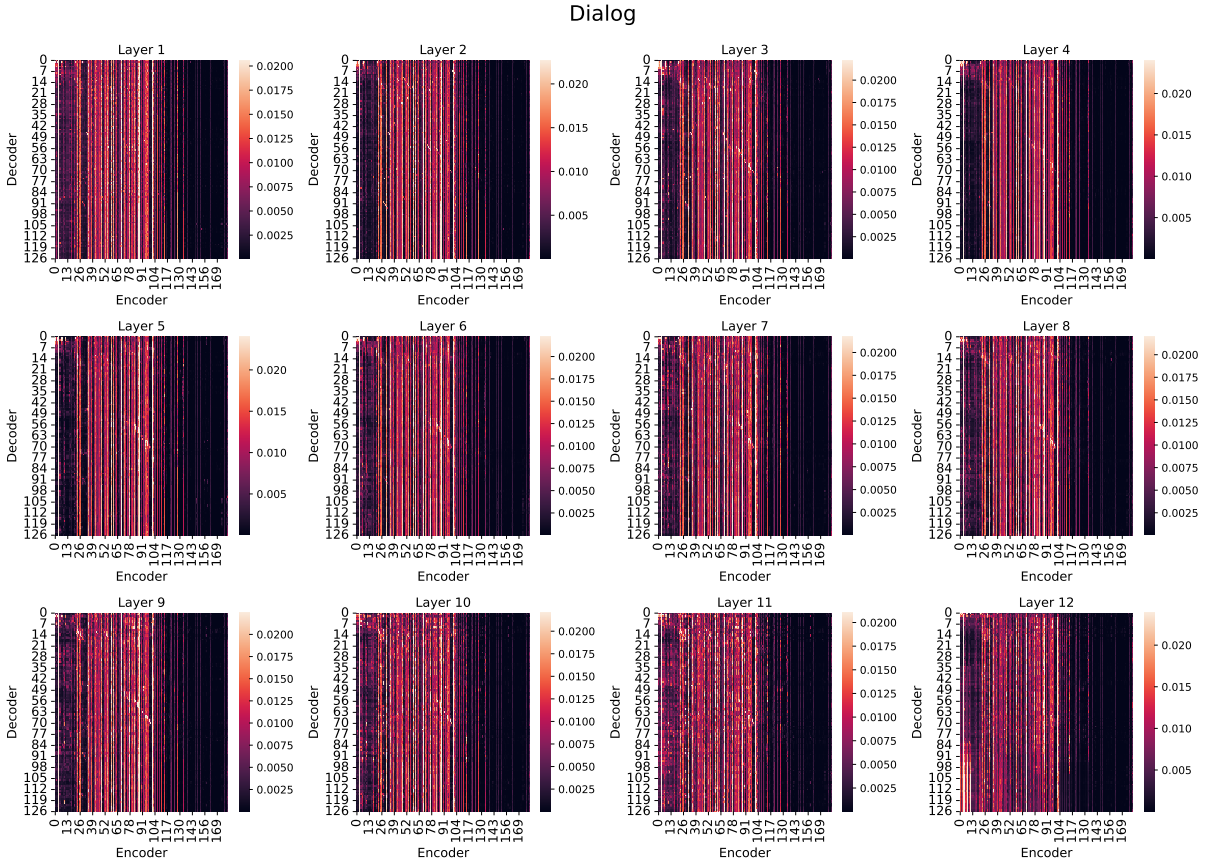


Figure 12: Average Decode-encoder attention heatmaps on dialog from different layers

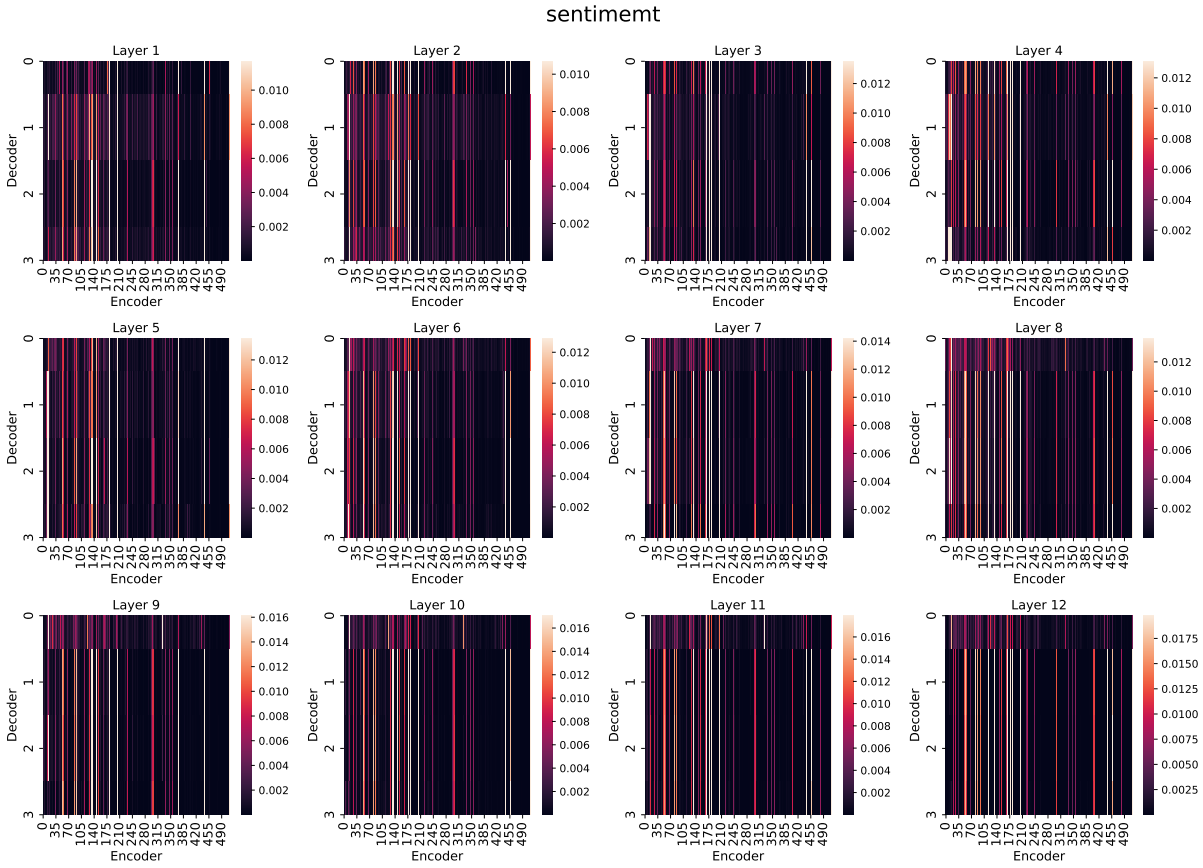


Figure 13: Average decode-encoder attention heatmaps on sentiment classification from different layers

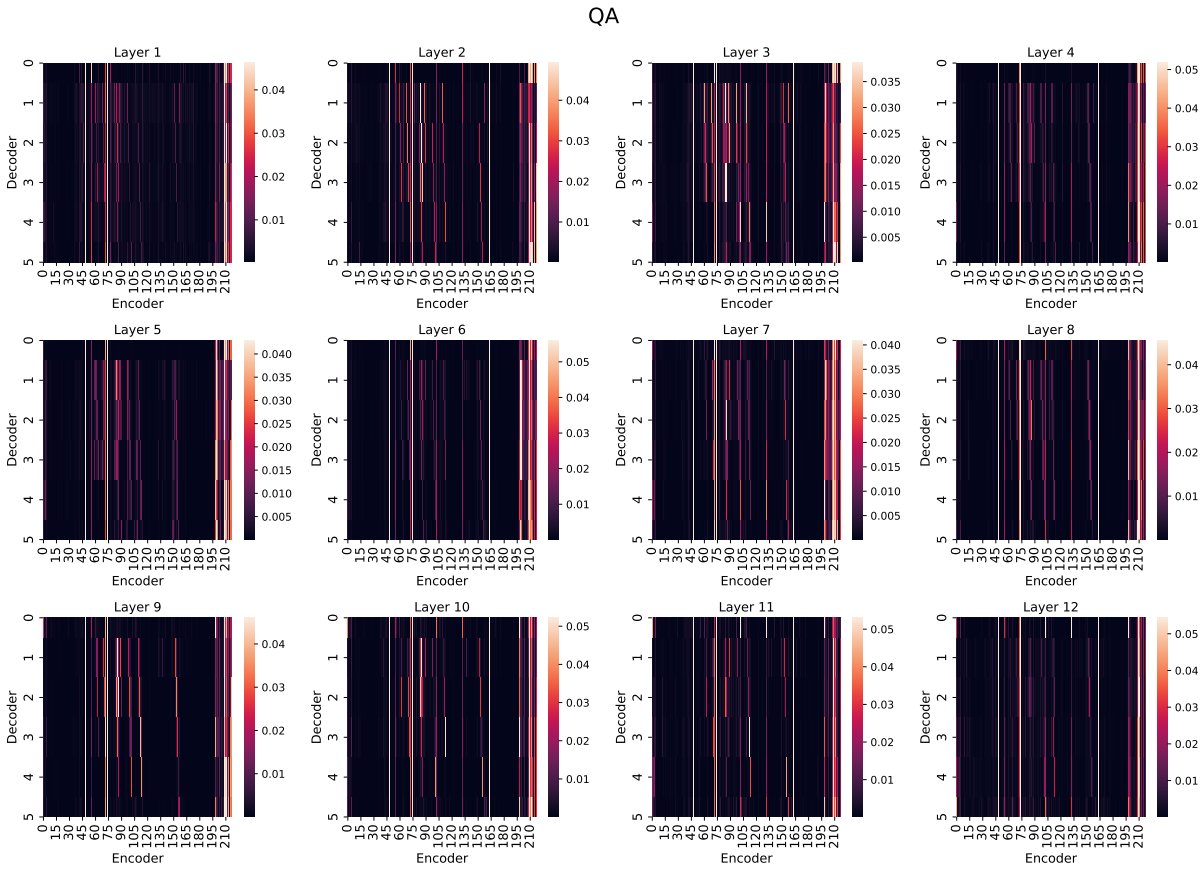


Figure 14: Average decode-encoder attention heatmaps on QA from different layers