Fine-tuned Language Models are Continual Learners

Thomas Scialom[1*] **Tuhin Chakrabarty**[2*] **Smaranda Muresan** [2]
[1]Meta AI
[2]Department of Computer Science, Columbia University
tscialom@fb.com, tuhin.chakr@cs.columbia.edu, smara@cs.columbia.edu

Abstract

Recent work on large language models relies on the intuition that most natural language processing tasks can be described via natural language instructions and that models trained on these instructions show strong zero-shot performance on several standard datasets. However, these models even though impressive still perform poorly on a wide range of tasks outside of their respective training and evaluation sets. To address this limitation, we argue that a model should be able to keep extending its knowledge and abilities, without forgetting previous skills. In spite of the limited success of Continual Learning we show that *Fine-tuned Language Models can be continual learners*. We empirically investigate the reason for this success and conclude that Continual Learning emerges from self-supervision pre-training. Our resulting model Continual-T0 (CT0) is able to learn 8 new diverse language generation tasks, while still maintaining good performance on previous tasks, spanning in total 70 datasets. Finally, we show that CT0 is able to combine instructions in ways it was never trained for, demonstrating some level of instruction compositionality.[1]

1 Introduction

Recent work has shown that large language models have the ability to perform zero-shot and few-shot learning reasonably well (Brown et al., 2020; Rae et al., 2021; Smith et al., 2022). A particularly successful line of work relies on the intuition that most natural language processing tasks can be described via natural language instructions. For example, a summarization task can be reformatted as a response to a natural language input as shown in Table 1. Sanh et al. (2022) and Wei et al. (2022) have released T0 and FLAN respectively and shown that fine-tuning models on a massive mixture of NLP datasets expressed via such natural language instructions (i.e., instruction tuning), improves the zero-shot performance of large language models. FLAN is extremely large in size (137B) and is not publicly available limiting its further use and reproducibility. Conversely T0 (Sanh et al., 2022) is publicly available and orders of magnitude smaller and hence we resort to working with T0.

The picture appeared on the wall of a Poundland store on Whymark Avenue [...] **How would you rephrase that in a few words?**
Graffiti artist Banksy is believed to be behind [....]

Table 1: An instance from T0 training set (Sanh et al., 2022) where a summarization task is reformatted as a natural language response to a natural language input.

However impressive, these models are still limited to simple instructions and mainly Natural Language Understanding (NLU) tasks. These models perform poorly on a wide range of tasks that differ from their respective evaluation sets. To improve their ability on new and diverse tasks, one needs to fine-tune these models again. However, one key problem associated with fine-tuning is *catastrophic forgetting* (French, 1999). So, how can we extend these models knowledge and abilities, without suffering from catastrophic forgetting?

In this paper, we study Continual Learning of large language models *fine-tuned on natural language instructions* and investigate their ability to adapt to diverse tasks, while avoiding catastrophic forgetting on the older tasks. For this purpose, we propose Continual-T0 (CT0), a T0 model that uses Continual Learning with rehearsal (Shin et al., 2017), i.e. using a memory buffer containing a small portion of previous data replayed during training (Section 3). We start from T0, a model trained jointly on 50 datasets, resulting in a good zero-shot performance on 12 completely different datasets. We are then able to teach progressively 8 new diverse tasks, while maintaining almost 100% of the

*Both Authors Contributed Equally
[1]Our code is publicly available at https://github.com/ThomasScialom/T0_continual_learning.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6107–6122
December 7-11, 2022

initial performance on all the previous datasets. This result is obtained by using only 1% of data for memory buffer. Notably, we also maintain the performance for the T0 zero-shot evaluation datasets, even though no rehearsal for those was done, the first of its kind setup for CL (Section 4).

Our final model, Continual-T0 (CT0) in addition to performing as well as T0 on all the different T0 datasets, can also understand instructions about the newly introduced tasks focused on language generation problems such as writing a haiku, generating empathetic responses in a dialogue, simplifying text, generating a headline with decoding constraints, generating natural language explanations for Natural Language Inferece (NLI) tasks, generating a Tweet on a given topic in the style of a given author, or question answering for new domain/concepts such as COVID-19.

We also conduct an extensive analysis and show that our newly learned instructions can be composed in ways never seen during training, leading to better generalization (Section 4.3). Given the surprising performance of a simple continual learning strategy, we empirically investigate the reason for this success. Why transformer models like T0 are continual learners? Is it because of their multi-task nature or the instruction tuning paradigm? Or does the large scale parameterization of language models contribute to this success? Our experimental analysis show that the easy adaptability and continual learning capabilities actually emerge from pre-training and not the above, **including scale** (Table 5, Section 5.1).

2 Related Work

Continual Learning Current models are limited in continuously learning without forgetting any previously acquired knowledge and abilities. Research in this direction has investigated various strategies such as External Memory, Constraints and Model Plasticity (Parisi et al., 2019). External Memory methods often simply use rehearsal with a replay during training (Rebuffi et al., 2017). de Masson D'Autume et al. (2019) also proposed local fine-tuning at inference time, leveraging examples similar to the considered input.

Through the lens of NLP tasks, Biesialska et al. (2020) look at the problem of CL and discuss major challenges involved. Jin et al. (2021) show CL algorithms are effective for knowledge preservation. Their study also infer that continual pretraining improves temporal generalization. (Douillard et al., 2021) proposed a a dynamic expansion of special tokens with a transformer architecture. Mi et al. (2020) and Madotto et al. (2021) perform CL for task oriented dialog systems by using replay based strategy. Cao et al. (2021) propose a new CL framework for NMT models, while Ke et al. (2021) proposes a novel capsule network based model called B-CL (Bert based CL) for sentiment classification tasks. Jin et al. (2020) show how existing CL algorithms fail at learning compositional phrases. Lin et al. (2022) propose a benchmark and highlight key challenges for continual model refinement in Out-of-Distribution data streams. More recently, Sun et al. (2019) propose a lifelong learning method LAMOL that is capable of continually learning new tasks by replaying pseudo-samples of previous tasks that require no extra memory or model capacity. To the best of our knowledge, LAMOL corresponds to the state-of-the-art for CL in NLP. Most similar to our work is that of Yin et al. (2022) who also study continual learning from task instructions based on the NATURAL-INSTRUCTION benchmark (Mishra et al., 2022).Finally instead of limiting to vision-only and language-only tasks Srinivasan et al. (2022) study the challenge of learning multimodal tasks in a CL setting, and systematically evaluate how upstream continual learning can rapidly generalize to new multimodal and unimodal tasks

Most of the aforementioned works fall into the 2 scenarios differentiated by Lomonaco and Maltoni (2017): 1) learning new data of known classes (online learning), and 2) learning new classes (class-incremental learning). Thus, the study are often limited to a narrow domain, or a specific task. In our work, we propose to address Continual Learning more broadly: learning a diverse set of new tasks different from the ones used for training. For this, we leverage the idea of instruction tuning (Wei et al., 2022; Sanh et al., 2022), that enables us to frame any NLP task as a response to a natural language input and use rehearsal as a mechanism to avoid catastrophic forgetting (Shin et al., 2017).

3 Continual Learning for Fine-tuned Language Models

3.1 Continual Learning via Rehearsal (CLR)

Our objective is to maintain the model's existing learned skills, while progressively learning more tasks. To prevent the model from catastrophic for-

getting, we rely on an external memory module, storing a subset of previous training data (Shin et al., 2017). We define the tasks to be learned as a task sequence $T = (T_1, T_2, , T_N)$ of N tasks. D_i is the corresponding dataset for task T_i. Formally, the training data augmented with **r**ehearsal D_i^r is defined as:

$$D_i^r = D_i \bigcup \sum_{j=1}^{i-1} (rD_j) \quad (1)$$

where r is the rehearsal hyper-parameter that controls the percentage of examples sampled from previous tasks $T_1, ...T_{i-1}$. We note that $r = 0$ corresponds to no memory, and $r = 1$ is equivalent to a multi-task setup using all the previous examples.

3.2 Continual-T0 (CT0)

For all our experiments, we instantiate our model with the T0 model (Sanh et al., 2022). T0 is a T5 model (Raffel et al., 2020) fine-tuned in a multitask setting on 50 datasets, where the natural language instructions corresponding to individual tasks are used as the input. The set of these 50 tasks corresponds therefore to T_1 in 1. This massive instruction tuning allows the model to perform well in a zero-shot setup, by leveraging the information presents only in the instructions. Our initial model is T0_3B, the T0 version with (only) 3 Billions parameters for all our experiments. We used the same hyper-parameters as the ones reported in Sanh et al. (2022)[2]. The only new hyper-parameter introduced in our paper is the *rehearsal proportion* r. We explore $r \in [0, 0.25\%, 1\%]$ as reported in our first set of results (see Section 3).

For each of T0 training tasks,we consider 100,000 examples for training, such that 1% rehearsal corresponds to 1,000 examples that will be used as the memory buffer for rehearsal. Thus, for datasets with fewer training examples, we upsample them and conversely for largest datasets like Gigaword or Simplification, we limit to 100,000 examples. Note that here, **while we used rehearsal for the *training* data of T0 training tasks, we never used any data from T0 zeroshot tasks, so it remains completely zero-shot**. It is important to highlight that rehearsal is the standard for CL, and a zero-shot set up with no rehearsal has never been explored yet to the best of our knowledge.

[2]See more details at https://huggingface.co/bigscience/T0pp

3.3 Tasks

We briefly describe all the tasks T used to progressively train and evaluate our model (a more complete description is also given in Appendix 7.2).

T0 Tasks. As detailed in Section 1, we instantiate our model with T0 weights. T0 is trained in a multitask setting on a collection of 50 datasets spanning from QA, Classification to Summarization. We refer to this set of 50 datasets as *T0 train (T0tr)*. To evaluate the true zero-shot performance for T0, the authors evaluated it on a set of 12 datasets corresponding to 4 tasks different from *T0 train*: Natural Language Inference, Co-reference resolution, Word sense disambiguation and Sentence completion. We refer to this set as *T0 zero-shot (T0zs)*.

New Tasks. To extend T0 capabilities and benchmark its performance in our continual learning setup,we introduce 8 new tasks focused on language generation, unlike the existing T0 evaluation tasks and majority of the T0 training tasks (except summarization). These tasks include: 1) **Text Simplification (Simpl)** with the goal of paraphrasing a given text using simple language, where we train our model on WikiAuto Jiang et al. (2020) and evaluate it on the WikiAuto and ASSET datasets (Alva-Manchego et al., 2020); 2) **Headline Generation with Constraint (HGen)**, where given a news article D and an input keyword X, the goal is to generate a headline that contains the keyword at the beginning, at the end or anywhere (see Table 2 for a sample instruction to generate a headline containing the keyword at the beginning). To create the training data, we simply leverage the gold-reference to select the keyword X, such that our model is trained with consistent and plausible instructions; 3) **Haiku Generation (Haiku)**, where the task is to generate a Haiku — a type of short form poetry originally from Japan — given a topic (see Table 2 for a sample instruction). We train on pairs (Haiku, title) from Reddit and generate Haikus for novel topics at inference time; 4) **Covid QA (CQA)** (Möller et al., 2020), a Question answering task focusing on COVID-19. Because T0 has been extensively trained on a QA dataset, CovidQA in its original format simply requires domain transfer. To make the task more challenging, we propose to provide only the question as an input, now framing the task as "learn the answer by heart" in an encyclopedia style task. This way the task framing can be seen as a new strategy to incorpo-

Text Smpfl(Simp)	Instruction	Make this text simpler: "A Georgian inscription around the drum attests his name."
	Output	*A Georgian writing on the drum is his name.*
Headline Generation (HGen)	Instruction	Make a title for this article that begins with "protesters": police arrested five anti-nuclear protesters thursday after they sought to disrupt loading of a french antarctic research and supply vessel , a spokesman for the protesters said .
	Output	*protesters target french research ship*
Haiku Gen (Haiku)	Instruction	Generate a haiku about 'Seagulls crying high'
	Output	*Seagulls crying high / the air smelling of sea salt / Or is it my tears?*
Covid QA (CQA)	Instruction	In the context of the COVID pandemic,who is at greater risk of dying from COVID19?
	Output	*patients with underlying medical conditions and the elderly*
Inquisitive Question Gen(InqQG)	Instruction	Given the following text, write the possible curious question it answers: "Positrons do not travel backwards in time. Positron-electron annihilation ..... So, we know they collide frequently enough to light up the galaxy in that part of the spectrum
	Output	*How often do electrons and positrons collide in nature?*
Empathetic Dialog Generation (EmDg)	Instruction	The associated emotion is "disappointed" and the input prompt is "Had to cancel our family vacation coming up next month. My husband's work said he couldn't go after they already approved the time off.". Now what would be your response, given the following dialogue context:=== - I had to cancel our family vacation coming up next month.
	Output	*I am really sorry to hear that. I hope everything is alright.*
Explanation Generation (Exp)	Instruction	Explain why the two following sentences are contradicting each other: "Sentence 1: A statue at a museum that no seems to be looking at."; Sentence 2: "Tons of people are gathered around the statue."
	Output	*If tons of people are gathered around the statue, it is not possible that no one seems to be looking at it.*
Twitter Stylometry (TwSt)	Instruction	Write a tweet about #WelcomeToNewYork, in the style of taylorswift13
	Output	*GUYS. #WelcomeToNewYork will be up on iTunes any minute now. This is not a drill!! GO GO GO*

Table 2: Example Instructions with their respective ground-truth for 8 new tasks learned continually from T0.

rating knowledge and preventing the model from concept drift.

5) **Inquisitive Question Generation (InqQG)** where we train our model on the ELI5 dataset (Fan et al., 2019) to generate questions that typically require long form answers; 6) **Empathetic Dialogue Generation (EmDg)**, where we generate a response to a conversational context grounded in emotional situations using the Empathetic Dialogue data Rashkin et al. (2019); 7) **Explanation Generation (Exp)** where we train a model on the eSNLI (Camburu et al., 2018) benchmark to generate natural language explanations given a premise, hypothesis and a label (entail, contradict, neutral); 8) **Twitter Stylometry (TwSt)**, where we generate a relevant tweet given a hashtag and the tweet's author by fine-tuning on the data consisting of tweets from the top 20 most followed users in Twitter released by Tareaf (2017). We illustrate the 8 new tasks with their instructions in Table 2. A complete detailed description for all the 8 tasks with train, validation splits is available in the Appendix 7.3.1.

3.4 Automatic Metrics

We report the accuracy for T0 zero-shot tasks, and standard metrics for NLG like BLEU(Papineni et al., 2002) and SARI(Xu et al., 2016) for Simplification, ROUGE (Lin, 2004) for Headline Generation, or BERTScore (Zhang et al., 2020) [3]for open-domain NLG tasks as it has been found to correlate well with human judgements.

We also designed customized metrics for some of the tasks.[4] For instance, to evaluate Twitter Stylometry where the task is to generate a tweet in the style of the author, we trained a Ridge Classifier to predict the author given the evaluated tweet. For Haiku generation, we know that in general, a Haiku contains only 17 syllables, broken up into three lines. We therefore create a metric to reflect the task structure that integrates i) the differences in syllables and number of lines between the gold and generated haiku, ii) the BLEU score between gold

[3]We use the BERTScore version based off deberta-mnli

[4]All those metrics implementations are available in the publicly released code.

and predicted, and iii) the presence of the topic in the generated haiku. We report all the details for the metrics in the Appendix 7.4.

4 Results

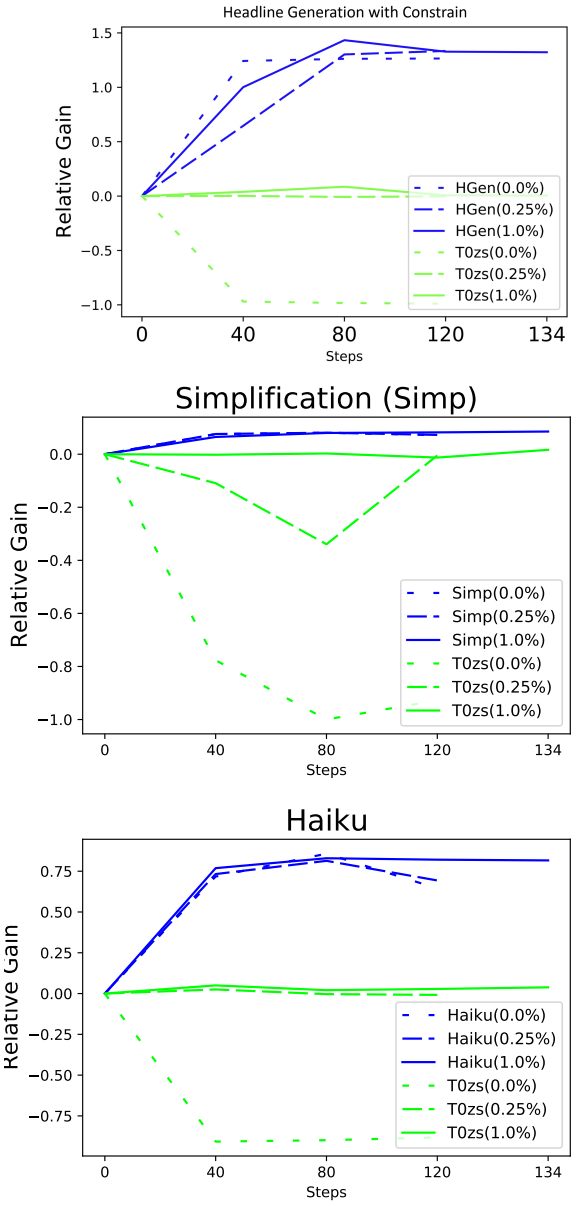

Figure 1: Rehearsal ablation with 0.0, 0.25 and 1.0% of training data showing target task performance along with T0 zero-shot performance(T0zs) with Relative Gain in Y axis vs Number of training steps in X axis. The results are normalised in % such that -1 corresponds to 100% decrease and +1 means +100% increase w.r.t. the initial performance.

4.1 Learning a New Task at a time

First, we test CLR independently on three tasks (Headline Generation with Constraint, Simplification, and Haiku Generation), by varying the rehearsal hyper-parameter between 0%, 0.25% and 1%, respectively. We report the results in terms of *Relative Gain* in Figure 1.

We observe that for the three tasks, the rehearsal value does not affect the task result: all the blue curves are consistent. Conversely, the rehearsal value has a dramatic impact on the T0 zero-shot results (green curves). As already discussed previously, at 0% rehearsal,the model catastrophically forgets the T0 zero-shot tasks. Conversely, with only 0.25% rehearsal we observe an almost perfect stability. Finally, with 1% rehearsal (solid line), T0 zero-shot results are stationary, indicating that our model is able to maintain its performance on those tasks, while learning a new task.

4.2 Learning a Sequence of New Tasks

As observed from our previous experiments using Continual Learning via rehearsal we can learn a new task at any time without catastrophic forgetting, with just a very little rehearsal percentage. As a next step, we propose to measure whether language models can progressively learn more tasks without catastrophic forgetting. This is an important direction as it would allow the models to continually increase their knowledge and capabilities without forgetting the knowledge already acquired.

To test this hypothesis, we start from T0 checkpoint, a model trained on 50 datasets. We progressively train it on a sequence of 8 new NLG tasks (see Section 7.3.1 and Table 2 for description of those tasks) using Continual Learning via rehearsal ($r = 1\%$). We call our final model CT0.

To measure the actual success for CL on a sequence of N tasks, we introduce the notion of *Upper Bound* (UB). UB corresponds to the maximum performance achieved by the model, when fine-tuned only on a specific task, T_n. Arguably, the model succeeds in CL, if it maintains a performance close to UB, while learning new tasks. The normalised results, i.e .,*Relative Gain* for a given task T_n, correspond to the actual scores s divided by their task T_n UB, s_{T_n}/UB_{T_n}. Hence, 1 corresponds to performing similar to the UB for any task. The model is expected to start bellow 1 before step n since it has not been trained yet on T_n, while for the latest steps t with $t > n$, results below 1 indicate task forgetting.

In Figure 2, we display the progressive sequential learning on the 8 new tasks. We learn a new task, starting from T0, and add to our rehearsal

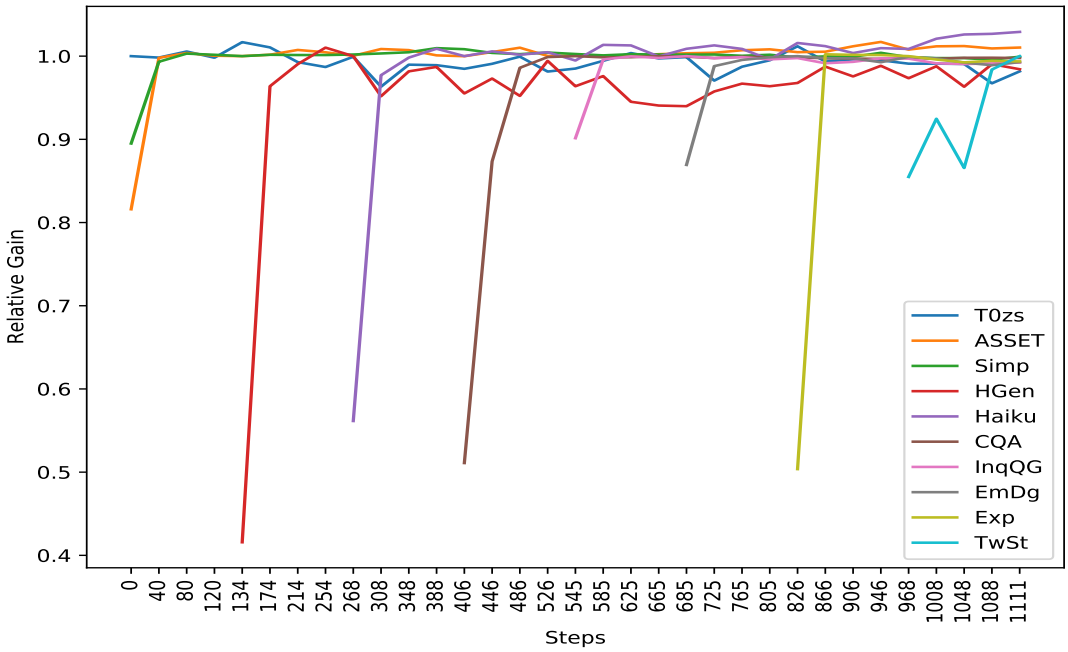

Figure 2: Progressive Relative Gain results for CT0 (11B) during the sequential learning(Y axis) vs Number of Training steps(X axis). The curves for tasks $T_0, ...T_7$ are displayed respectively at step $0, ..., i$ such that only the first task, Simplification (green and orange) is present at step 0, then HGen (red) etc.

buffer 1% of the data of the learned task. We observe an improvement progressively for each task, that is our model keeps learning new tasks. At the same time, the performance is preserved for the other tasks, (i.e., the Relative gain remains around 1) indicating the success of our CLR method in a sequential learning setup through more than 1000 gradient steps over 8 different tasks.

In Table 3, we report the results on all the 8 new tasks as well as T0tr and T0zs (see Section 3.3), corresponding respectively to the evaluation sets of the 50 training datasets used in T0, and the 12 datasets kept apart for the zero-shot evaluation. In the first bloc of Table 3, we observe the starting performance of our two initial checkpoints, T0_3B and T0pp(11B). The second bloc corresponds to their respective Upper Bounds. We report the results for our models after training them progressively on the 8 new tasks, as well as the baseline LAMOL (see Section 2; for fair comparison we adapted LAMOL initialising it with T03B, additional details can be found in Appendix 7). The CT03B and CT0pp results in Table 3 are reported after the model was fine-tuned on the latest task in the sequence (intermediary steps are given in Table 7 in Appendix).

Our two CT0 models obtain final results very close to their UB, maintaining 99.8% for T0pp and 98.0% for T0_3B. This clearly indicates the efficiency of the CLR method. Notably, no task suffers a decrease in performance more than 2% for T0pp. Table 3 shows how the CT0 model remembers and retains knowledge from tasks trained at very early stages of the Continual Learning process. Moreover, CT0 still performs well on the zero-shot set of tasks (T0zs) despite no rehearsal for those.

It should also be noted that the T0pp model fails to generalize for most NLG tasks, as opposed to our CT0 model. For instance Table 6 in Appendix shows it can generate a haiku that has a perfect syllable count of 17 given an unseen topic of 'mountain winds haunt'. It can also generate reasonable natural language explanations that often comply with our commonsense. Moreover, CT0 obtains a new state-of-the-art on the ASSET evaluation set, improving over MUSS (Martin et al., 2020): 85.9 BLEU4 Vs 72.98 and 46.6 SARI Vs 44.15, and despite not using all the training data available.

In contrast to Continual Learning with rehearsal, LAMOL clearly diverges from its UB (T03B) indicating catastrophic forgetting. While LAMOL was known to perform well mostly on NLU tasks, we hypothesise that the generative nature for our tasks is not suited for the method. Finally, **Continual Learning with rehearsal approach is *task order invariant*** as demonstrated by *revfinal* results: *revfinal* corresponds to CT03B trained on the 8 tasks within in the reverse order [5]. We give more details

[5]We report task order invariance results only using 3B and

about the order choice in the Appendix.

4.3 Zero-shot Instruction Compositionality

Our CT0 model has learned effectively to process different instructions in specific contexts: word level constraint in the context of headline generation, or an emotional tone in the context of dialogue. Does CT0 understand these instructions in different contexts? To answer this question, and to explore whether CT0 can learn instruction compositionality we conduct several experiments.

Zero-Shot Constraint. In Table 4 we explore how our model succeeds in understanding constraint instructions beyond the one it was exposed during training. Our model was trained on Headline Generation with Constraint (HGen) instructions with only one match, such as *Make a title for this article containing "X"*. To test generalization, we prompt our CT0 model with unseen instructions with 2 and 3 matches, such as *Make a title for this article containing "X" and "Y"*, or *Make a title for this article containing "X" and "Y" and "Z"*. We also compose instructions from constraint and Twitter Stylometry resulting in instructions such as *Write a tweet about X, in the style of Y, containing Z*. CT0 respects the *Contain* constraint 77% for $n = 1$. The score naturally drops when $n > 1$, however the satisfiablity is still 50% of the time for $n = 2$ and 40% for $n = 3$. As expected, the ROUGE-1 score also improves: *NoCons*: 30.2, #Cons=1: 38.9, #Cons=2: 43.9 and #Cons=3: 47.4. When we compose HGen and TwSt, CT0 also performs significantly better compared to $CT0_{NoCons}$ (46.4 vs. 10.7).

Zero-Shot Emotional Haiku. We explore whether combining an emotion with the Haiku instructions would help control the haiku generation. Note that during training, only the task of Empathetic Dialogue has been exposed to emotion. Our results, reported in Figure 3, indicate that CT0 is able to combine an emotion with the Haiku instructions in a zero-shot setting. For instance, given the following new instruction *Generate a haiku about "held my hand". The associated emotion is "faithful".*, our model output is "*He held my hand through thick and thin, Through sickness and health, through life and death*". A qualitative analysis also shows that CT0 understands subtle nuances; for instance given as input *Generate a haiku about "Seagulls crying high". The associated emotion is "nostalgic".* our model output is "*Seagulls crying high, A familiar scene, from a childhood Now* ".

not 11B due to computing restrictions

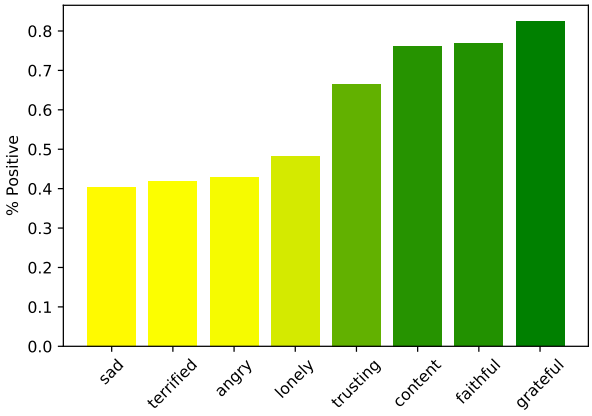

Figure 3: Emotion Generalization: Percentage of Haiku classified as positive, when adding emotion specific constraints to the Haiku instruction. We used an open source binary sentiment analysis classifier.[6]

These are promising results regarding CT0's ability to comprehend new instructions, including instruction composition. While contemporaneous work by Nayak et al. (2022) propose a novel form of soft prompting for compositional zero-shot learning we show that a continually fine-tuned language model is able to perform the same.

5 Discussion

5.1 Why could LLMs be lifelong learners?

Given our current experimental protocol, one can draw different hypotheses: is CL a consequence emerging from the massive multi-task pre-training in T0? Or from the instruction tuning paradigm of T0? Or from the scaling law as studied by Ramasesh et al. (2021)? To answer this research question, we applied the same CL setup starting from 1) T5-small, 2) T5-3B, and 3) a T5-3B architecture randomly initialised. Our results in Table 5 show that CT**5** with 3B parameters performs similar to CT**0**3B on the 8 tasks. While CT**5**-small obtains as expected a lower average performance, it still mostly maintains great results w.r.t. its Upper Bound, indicating that CL does not emerge from scale. Conversely, when initialised randomly the model is not even able to obtain a good UB. These results draw a clear conclusions: **CL emerges from the intensive pre-training stage**. This confirms contemporaneous findings by Cossu et al. (2022)

	T0tr R1	T0zs Acc	ASSET B4/SARI	Simp B4/SARI	HGen R1/Cons	Haiku H_{cust}	CQA BS	InqQG 1Tok/BS	EmDg BS	Exp BS	TwSt Clf/BS
T0_3B	49.8	48.2	70.1/41.0	12.8/41.1	33.6/32.2	34.2	47.6	2.1/58.7	48.6	32.7	54.4/38.0
T0pp	54.2	65.6	56.5/37.7	11.7/40.1	34.9/35.9	31.6	46.0	2.4/59.8	49.7	37.2	66.4/45.1
UB_3B	49.8	48.2	79.9/45.2	13.8/44.6	39.7/81.0	62.6	90.0	5.3/63.3	55.7	71.8	74.8/56.5
UB_pp	54.2	65.6	85.3/46.1	15.0/44.8	41.9/86.9	63.9	90.0	4.9/65.7	56.6	73.5	74.4/57.9
Lamol	32.6	33.6	37.3/12.6	8.4/21.4	22.9/33.5	25.8	46.6	1.8/47.9	45.1	27.6	50.1/35.2
CT03B	47.9	46.6	78.0/44.5	14.6/43.7	37.3/77.5	60.4	86.8	5.2/61.9	55.3	72.4	74.8/56.5
CT0pp	**53.7**	**64.4**	**85.9/46.6**	**14.6/44.7**	**40.7/85.5**	**65.8**	**89.8**	**4.8/65.2**	**56.2**	**73.0**	**74.4/57.9**
revfinal	48.1	48.8	83.3/45.4	14.6/43.9	39.0/81.6	61.2	88.6	4.4/61.9	55.0	72.4	73.2/57.3

Table 3: Results for the starting checkpoints T0_3B and T0pp(11B), their upper bounds scores and our final models as well as LAMOL. Bolded result means there less than 2% forgetting. T0tr and T0zs denote respectively for T0 train-tasks and T0 zero-shot-tasks and are the average accuracy obtained on all their evaluation datasets. B4, R1, BS denote BLEU4, ROUGE-1 and BERTScore. Note that we detail the intermediate steps results in the Appendix.

	HGen			TwSt
# Cons	1	2	3	1
CT0	77.0	56.4	39.5	46.4
$CT0_{NoCons}$	33.6	15.4	8.1	10.7

Table 4: Table showing Constraint generalisation i.e % of instructions completely respected, when providing constraints for unseen prompts. $CT0_{NoCons}$ corresponds to providing the same input without constrain.

and Mehta et al. (2021) in other setups and even modalities. We report the detailed results for those experiments in the Appendix.

5.2 Toward Concept Drift

In the original CovidQA the task consists of answering a question present in a given paragraph. In this setup, one can arguably succeed into answering questions about COVID by transferring the task knowledge, even without particular domain knowledge about COVID. In our paper, we intentionally chose to not provide the context for CQA but only the question. This alternative setup corresponds to learning by heart the answer to a question. Our results in Table 3 show that while we framed CQA as a new task to learn, our proposed setup also opens new way to tackle concept drift, by directly incorporating knowledge into a model.

5.3 Data Efficiency

Our method based on rehearsal learning is simple yet efficient. While the complexity in term of data storage and training is not constant (O(1)), with only 1% of the previous training data we are able to retain model abilities. This result is still data and computationally efficient, compared to the standard approach of retraining the model from scratch on all tasks. In cases where the number of tasks to learn would grow by several order of magnitude, more sophisticated methods could be explored. We leave this for future research.

6 Conclusion

We explored for the first time Continual Learning for instruction-based models. Our results indicate that *fine-tuned Language Models are efficient continual learners*: 1% rehearsal is enough to maintain a high performance on previously learned tasks, while learning new ones. Additionally, we show that our model CT0 is able to comprehend new instructions obtained via instruction composition. The current technique to learn multiple tasks is to train a model from scratch. We hope this work paves the way toward a new paradigm where models do not have to be retrained all over again. We believe our experimental findings will contribute to the effectiveness of large language models, enabling them to progressively adapt to new concepts and acquire more and more abilities. As an analogy with Software Development, this could be seen as *learning* new features. New checkpoints are like new versions of a model. In this context, Continual Learning will help toward the *Call to Build Models Like We Build Open-Source Software*.[7]

[7] https://tinyurl.com/3b7b2nrc

	T0tr R1	T0zs Acc	ASSET B4/SARI	Simp B4/SARI	HGen R1/Cons	Haiku H_{cust}	CQA BS	InqQG 1Tok/BS	EmDg BS	Exp BS	TwSt Clf/BS
UB_rand	N/A	N/A	0.5/24.3	0.0/29.6	1.5/0.1	9.6	25.2	1.2/25.4	36.3	33.1	24.7
UB_T5small	N/A	N/A	87.8/45.9	15.6/43.2	35.3/67.8	53.4	54.1	3.4/57.0	51.3	33.8	52.4/54.6
UB_T53b	N/A	N/A	87.0/45.6	15.4/43.7	33.0/89.4	63.0	89.9	2.92/61.5	55.3	71.6	75.6/55.4
UB_T0	49.8	48.2	79.9/45.2	13.8/44.6	39.7/81.0	62.6	90.0	5.3/63.3	55.7	71.8	74.8/56.5
CTrand	N/A	N/A	0.0/22.9	0.0/28.5	0.2/0.0	9.6	25.2	1.2/27.9	28.1	30.7	24.7
CT5small	N/A	N/A	85.5/45.8	15.0/42.8	34.6/64.8	51.8	49.5	3.3/56.0	51.2	32.3	52.4/54.6
CT53B	N/A	N/A	84.6/45.8	14.8/44.0	38.3/88.3	62.3	85.8	4.64/62.1	55.5	73.1	75.6/55.4
CT03B	47.9	46.6	78.0/44.5	14.6/43.7	37.3/77.5	60.4	86.8	5.2/61.9	55.3	72.4	74.8/56.5

Table 5: Results including T5-small and T5-3B, T0_3B, and a 3B Transformer randomly initialised. We can observe that 1) only CTrand largely degrades w.r.t. its UB, UB_rand; 2) even T5_small is able to mostly maintain its performance indicating that scale is not what matter the most.

Instr	*Make a title for this article, finishing with "escalates":* the sri lankan government announced the closure of government schools with immediate effect as a military campaign against tamil separatists escalated in the north of the country .
CT0	sri lanka closes schools as war with tamils **escalates**
T0pp	sri lanka closes schools as tamil rebels advance
Instr	*Write a haiku about* 'mountain winds haunt'
CT0	**mountain winds haunt**, the hollow of the stones, voices echo there.
T0pp	a lone tree in the mountains is haunted by the wind
Instr	*Explain why the two following sentences do not entail each other:* "Sentence 1: A woman with a green headscarf, blue shirt and a very big grin."; Sentence 2:"The woman has been shot."
CT0	A woman cannot be smiling if she has been shot.
T0pp	No

Table 6: Outputs for HGen, Haiku and Exp from T0pp and our continually learned final model CT0.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. Tuhin is funded by Columbia Center of Artifical Intelligence & Technology (CAIT) and the Amazon Science Ph.D. Fellowship).

Limitations

As discussed in 5.3, CL with rehearsal still requires to use a buffer of data previously seen which limits several scenarios where those data would not be available anymore. While we have done our best to select numerous and diverse tasks in this paper, it still represents a limited set. Would our results still hold given hundred or thousand tasks? In other modalities? It should also be noted that our study is limited to English-only datasets, as we started from T0 which is not multilingual in nature. Additionally while results using automatic metrics give a fair idea of task performance and measuring CL abilities, we would like to conduct a human evaluation in near future although its expensive give the size of test data and the number of tasks

Ethics Statement

Although we use language models trained on data collected from the Web, which have been shown to have issues with gender bias and abusive language, the inductive bias of our models should limit inadvertent negative impacts. Unlike model variants such as GPT, T5 is a conditional language model, which provides more control of the generated output. We have verified carefully that our training or evaluation data does not contain any toxic text and it underwent manual inspection by the authors and experts. We also believe our work in continual learning is a step towards data efficiency and conservation of computing resources, as one saves training time by only using 1% rehearsal

References

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational*

Linguistics, pages 4668–4679, Online. Association for Computational Linguistics.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Yue Cao, Hao-Ran Wei, Boxing Chen, and Xiaojun Wan. 2021. Continual learning for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3964–3974, Online. Association for Computational Linguistics.

Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. 2022. Continual pre-training mitigates forgetting in language and vision. *arXiv preprint arXiv:2205.09357*.

Cyprien de Masson D'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, 32.

Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. 2021. Dytox: Transformers for continual learning with dynamic token expansion. *arXiv preprint arXiv:2111.11326*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7943–7960. Association for Computational Linguistics.

Xisen Jin, Junyi Du, Arka Sadhu, Ram Nevatia, and Xiang Ren. 2020. Visually grounded continual learning of compositional phrases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2018–2029, Online. Association for Computational Linguistics.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2021. Lifelong pretraining: Continually adapting language models to emerging corpora. *arXiv preprint arXiv:2110.08534*.

Zixuan Ke, Hu Xu, and Bing Liu. 2021. Adapting BERT for continual learning of a sequence of aspect sentiment classification tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4746–4755, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Sida Wang, Xi Lin, Robin Jia, Lin Xiao, Xiang Ren, and Scott Yih. 2022. On continual model refinement in out-of-distribution data streams. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3128–3139, Dublin, Ireland. Association for Computational Linguistics.

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

Vincenzo Lomonaco and Davide Maltoni. 2017. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning*, pages 17–26. PMLR.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. 2021. Continual learning in task-oriented dialogue systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7452–7467, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2020. Muss: multilingual unsupervised sentence simplification by mining paraphrases. *arXiv preprint arXiv:2005.00352*.

Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. 2021. An empirical investigation of the role of pre-training in lifelong learning. *arXiv preprint arXiv:2112.09153*.

Fei Mi, Liangwei Chen, Mengjie Zhao, Minlie Huang, and Boi Faltings. 2020. Continual learning for natural language generation in task-oriented dialog systems. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3461–3474, Online. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. 2020. COVID-QA: A question answering dataset for COVID-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

Nihal V Nayak, Peilin Yu, and Stephen H Bach. 2022. Learning to compose soft prompts for compositional zero-shot learning. *arXiv preprint arXiv:2204.03574*.

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 on Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, Pennsylvania. ACL.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

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. *Journal of Machine Learning Research*, 21:1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. 2021. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

Arij Riabi, Thomas Scialom, Rachel Keraron, Benoît Sagot, Djamé Seddah, and Jacopo Staiano. 2021. Synthetic data augmentation for zero-shot cross-lingual question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7016–7030.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Thomas Scialom and Jacopo Staiano. 2020. Ask to learn: A study on curiosity-driven question generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2224–2235.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Tejas Srinivasan, Ting-Yun Chang, Leticia Leonor Pinto Alva, Georgios Chochlakis, Mohammad Rostami, and Jesse Thomason. 2022. Climb: A continual learning benchmark for vision-and-language tasks. *arXiv preprint arXiv:2206.09059*.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. Lamol: Language modeling for lifelong language learning. In *International Conference on Learning Representations*.

Bin Tareaf. 2017. R.: Tweets dataset-top 20 most followed users in twitter social platform. *Harvard Dataverse*, 2.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing Statistical Machine Translation for Text Simplification.

Transactions of the Association for Computational Linguistics, 4:401–415.

Wenpeng Yin, Jia Li, and Caiming Xiong. 2022. ConTinTin: Continual learning from task instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3062–3072, Dublin, Ireland. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

7 Appendix

7.1 Tasks Order

The task order has been selected 1) randomly among the three first tasks Text Simplification, Headline Generation with Constraint and Haiku Generation, and 2) in light of the actual success, we progressively kept adding new tasks. This setup corresponds to a realistic usage of our proposed method, where future tasks were thus unknown even for us. To assess a potential impact of the order, we also conduct an alternative experiment with our 3B model, where the order is reversed. We did not experimented further different orders due to the high computation required.

7.2 Tasks

In this section, we describe all the tasks T used to progressively train and evaluate our model. For all the new tasks (i.e., not the T0 tasks), we also designed instructions, as illustrated in Table 2.

7.3 Automatic Metrics

7.3.1 New Tasks

All of our newly introduced tasks are language generation tasks in contrast to the T0 evaluation tasks and majority of the T0 training tasks (all except summarization).

Text Simplification (Simpl) Jiang et al. (2020) provided WikiAuto, a set of 400,000 aligned sentences from English Wikipedia and Simple English Wikipedia as a resource to train sentence simplification systems. The test set contains 4,000 examples. In addition, we also evaluate our models on a second Text Simplification dataset, ASSET (Alva-Manchego et al., 2020). This is a dataset dedicated for the evaluation of sentence simplification in English, providing 2,000 multiple references per example, unlike previous simplification datasets. Table 2 shows our designed instructions for this task.

Headline Generation with Constraint (HGen). While writing a title for a news article, it can be very useful to add additional constraints, such as the presence of certain words. However, traditional decoding strategies like the BeamSearch often fail to achieve this goal as discussed in 4. Gigaword is one of T0 training dataset. Our new task consists of generating a title given a news article *with additional constraints*. Towards this goal, for a given document D and an input keyword X we design the following three instructions: [*Make a title for this article,* ***starting with / ending with / that contains*** *"X" : D* where X is a word we want to be present in the output text at the beginning/end/anywhere, and D the source document, as illustrated in Table 2. To create the training data, we simply leverage the gold-reference to select the word X, such that our model is trained with consistent and plausible instructions. Gigaword contains millions of training examples. The original test set is composed of 1,951 examples, so we convert it to 3 sets of 1,951 examples for our Start/End/Contain instructions, respectively.

Haiku Generation (Haiku). For the task of haiku generation, we crawl[8] 10,718 haikus with at least 1 up-vote from the Subreddit haiku, [9] and split it in 9,742 and 974 example for the train and test sets, respectively. Table 2 shows an example instruction for Haiku Generation about a given topic.

Covid QA (CQA) Möller et al. (2020) created COVID-QA, a Question Answering dataset consisting of 2,019 question/answer pairs annotated by volunteer biomedical experts on scientific articles related to COVID-19. We consider this dataset since to the best of our knowledge, T0 has never been exposed to any COVID-19 related data. In its original version, the dataset is framed as SQuAD (Rajpurkar et al., 2016), with triplets (context, question, answer), where the context contains the answer. Because T0 has been extensively trained on QA dataset, CovidQA in its original format simply requires domain transfer. To make the task more challenging, we propose to provide only the question as an input, now framing the task as "learn the answer by heart" in an encyclopedia style task. This way the task framing can be seen as a new strategy to incorporating knowledge and preventing the model from concept drift.

Inquisitive Question Generation (InqQG) To foster long form question answering Fan et al. (2019) created the ELI5 dataset that comprises 270,000 English-language threads from the Reddit forum of the same name, [10] where an online community provides answers to several open ended inquisitive questions. Table 2 shows an example instruction in order to generate inquisitive ques-

[8] The crawling part was done by Tuhin Chakrabarty and at Columbia.

[9] https://www.reddit.com/r/haiku/

[10] https://www.reddit.com/r/ExplainLikeImfive/

tions. As opposed to standard Question Generation based on SQuAD, ELI5 enables open-ended questions, closer to human-style questions (Scialom and Staiano, 2020). We filtered out the Reddit threads to keep only well formed questions,[11] resulting in 61,710 and 1,681 examples for the training and test set, respectively.

Empathetic Dialogue Generation (EmDg) Rashkin et al. (2019) proposed a benchmark for empathetic dialogue generation by creating a dataset of conversations grounded in emotional situations. Each example in the dataset contains an input emotion, situation in which dialogue appears and the entire conversation. We display in Table 2 the corresponding instruction. At the example level, our training and test datasets contain 58,770 and 8,396 examples, respectively.

Explanation Generation (Exp). The Stanford Natural Language Inference dataset consists of a classification task, where given a Premise(P) and an Hypothesis(H), the model has to chose between 3 options: entailed, contradiction or not related. Camburu et al. (2018) extend this NLI dataset by annotating the explanations of the label in natural language. In our paper, we consider as input the Premise(P), the Hypothesis(H), and the label, and train our model to generate the explanation. The dataset is composed of 100,000 and 9,824 train and test examples, respectively.

Twitter Stylometry (TwSt) Tareaf (2017) extracted tweets from the top 20 most followed users in Twitter social platform, including singers such as Katy Perry or Selena Gomez, as well as the official account of Barack Obama when he was president of the USA. The style for tweets largely differs from one account to an another, e.g. @BarackObama: "*It's time to #ActOnClimate*" vs. @KimKardashian: "*makes me want to go back blonde but i'm scared it will ruin my hair :-(*". We define the Stylometry task as generating a relevant tweet given i) a hashtag, and ii) the tweet's author. We thus selected only tweets containing hashtags (#) from the original dataset, resulting in a total of 13,041 and 250 examples for train and test sets, respectively. We display at the bottom of Table 2 an example instruction for this task.

[11] I.e, starting in "W" or "H" and finishing with a question mark. See the code for the exact implementation, class ELI5promptFormat in data_handler.py.

7.4 Automatic Metrics

T0 zero-shot evaluation set (see Section 3.3) only contains tasks framed as classification. For T0 evaluation, Sanh et al. (2022) compute the log-likelihood of each of the target options, and the option with the highest log-likelihood is selected as the prediction. This strategy holds when restricting the evaluation to classification tasks. However, in the context of an open-ended model able to perform NLG tasks, a user is interested in the actual output of the model rather than probabilities. We therefore report the accuracy of the prediction compared to the ground-truth answer for all those tasks. This measure is more conservative, as it requires an exact match.

In the context of Continual Learning, we also suspect that using only a comparison of the log-likelihood of respective classes would not reflect the actual model's memory, since the decoders are known to suffer from catastrophic forgetting more than the encoders (Riabi et al., 2021).

Standard NLG Metrics. For the standard tasks, we rely on widely used metrics: ROUGE (Lin, 2004) for Summarization; BLEU (Papineni et al., 2002) and SARI (Xu et al., 2016) for Simplification. In this paper, we also include open-domain NLG tasks, such as Dialogue or Explanation generation. The space of possible correct outputs is too large in this case to rely on n-gram based metrics like BLEU or ROUGE. For this reason, we report BERTScore (Zhang et al., 2020) to measure the similarity between a prediction and its gold-reference in those tasks.[12]

When possible, we also designed customized metrics that are better suited for the task.[13]

Customized NLG Metrics.

- *Constraint*: For our prompts with *constraint*, such as "Write a text that *starts/contains/ends* with [some word]", we also report the accuracy of respecting the constraint. Concretely, an output is correct only if it contains the [word] at the right location: the beginning for *start*, the end for *end*; any location for *contain*.
- *First Word Distribution (1Tok).* In ELI5, the questions are supposed to be inquisitive, not factual like in SQuAD. Therefore, the distribution of the

[12] We used BERTScore based on *deberta-mnli* that is shown to have high correlation with human judgements.

[13] All those metrics implementations are available in the publicly released code.

first words is very informative. For instance, the percentage of questions starting with "why/how" is more important than "what". We therefore rely on the Jensen Shannon Divergence between the first words distributions of the ground truth examples and our predictions. We report its inverse, so the higher the better.

- *Author Classification (Clf)* In Twitter Stylometry, the author is part of the input, so the generated tweet is aligned with the author's style. To measure this condition, we train a classifier on the dataset, with the tweets as inputs, and the corresponding author names as target categories. We trained a Ridge Classifier using scikit-learn (Pedregosa et al., 2011), and obtained 0.81% accuracy. This high accuracy allows this Clf metric to be informative enough.
- H_{cust} Haiku is a type of short form poetry originally from Japan as illustrated in the Table 2. In general, it contains only 17 syllables, broken up into three lines. We calculate two differences between the prediction and the ground-truth: i) for the number of lines, and ii) for the number of syllables. H_{cust} corresponds to the average of these two differences, BLEU and the Constraint satisfiability (i.e., if the generated haiku contains the topic phrase X that was present in the instruction).

7.5 Evaluation for T0 Train Set

Because there are 50 datasets with thousands of examples in the test sets per task, evaluating on each examples would be computationally intensive. For this reason we restricted this set to 1000 examples randomly sampled from all the examples in the test sets. Because the set contains both NLG and NLU tasks, using the accuracy is not enough. For simplicity we used therefore ROUGE-1 which allows is consistent with accuracy for NLU tasks but also allows to take into account NLG evaluation,

7.6 Additional Results

In the main paper, Table 5 we reported the additional results when starting from T5 and a random transformer. These results are discussed in the first section of our Discussion.

In Table 7 we report the progressive results, and not just the initial checkpoint, the Upper Bound and the final model.

7.7 Implementation Details

For all our experiments woth T0_3B and T0pp, we instantiate our model with the T0 model (Sanh et al., 2022) using the official implementation. [14]

For fine-tuning T0_3B, we used the same hyper-parameters as the ones reported in Sanh et al. (2022): all the details from the batch-size to the learning rate are provided in details here. [15]

The only new hyper-parameter introduced in our paper is the *rehearsal proportion* r. We explored $r \in [0, 0.25\%, 1\%]$ as reported in our first set of results.

For each task, we consider 100,000 examples for training, such that 1% rehearsal corresponds to 1,000 examples from the memory buffer. Thus, for datasets with fewer training examples, we upsample them and conversely for largest datasets like Gigaword or Simplification, we limit to 100,000 examples.

When we scaled our best setup to the 11B parameters version of T0, *T0pp*, we observed instability in validation performance. Thus, we changed the learning rate from 1e-3 to 1e-4 as well as the optimizer to AdamW instead of Adafactor for all our 11B experiments. All the other hyper-parameters remain similar to the 3B model.

For the T5 ablations, we again used the Hugging Face implementations [16] and applied the same hyper-parameters as above.

At inference time, we use greedy decoding, i.e. a Beam Search with $K = 1$.

[14] https://huggingface.co/bigscience/T0pp
[15] https://huggingface.co/bigscience/T0pp
[16] https://huggingface.co/t5-3b

	T0zs Acc	ASSET B4/SARI	Simp B4/SARI	HGen R1/Cons	Haiku H_{cust}	CQA BS	InqQG 1Tok/BS	EmDg BS	Exp BS	TwSt Clf/BS
T0_3B	48.2	70.1/41.0	12.8/41.1	33.6/32.2	34.2	47.6	2.1/58.7	48.6	32.7	54.4/38.0
T0pp (11B)	65.6	56.5/37.7	11.7/40.1	34.9/35.9	31.6	46.0	2.4/59.8	49.7	37.2	66.4/45.1
+Simp 3B	48.9	79.9/45.2	13.8/44.6	30.3/31.0	30.9	43.9	2.0/56.1	40.2	34.9	50.8/42.5
+Simp 11B	**66.7**	85.3/46.1	15.0/44.8	34.9/36.1	33.0	47.2	2.1/59.0	48.1	39.2	68.8/47.6
+HGen 3B	46.9	81.4/44.9	14.1/43.9	39.7/81.0	33.7	44.2	2.5/55.9	45.9	55.2	19.6/37.3
+HGen 11B	65.5	84.5/46.1	**15.3**/44.8	**41.9/86.9**	35.9	46.6	2.9/59.7	48.9	36.4	69.6/48.1
+Haiku 3B	48.8	81.6/45.0	14.6/43.9	39.0/78.2	62.6	43.0	2.3/54.9	47.2	39.0	65.6/44.5
+Haiku 11B	64.6	83.5/46.1	14.9/45.1	41.1/83.0	63.9	46.0	2.9/59.9	48.9	37.5	66.4/46.2
+CQA 3B	48.5	79.7/44.4	14.0/43.8	37.6/75.4	62.2	90.0	2.0/54.4	42.5	38.7	66.4/45.3
+CQA 11B	64.6	84.3/46.1	14.5/**44.9**	40.9/83.7	63.6	**90.0**	2.9/59.2	48.5	42.7	67.2/47.3
+InqQG 3B	47.4	65.2/41.2	14.6/43.8	37.9/77.7	60.4	89.6	5.3/63.3	46.8	34.2	59.2/45.4
+InqQG 11B	65.5	85.5/46.3	14.9/44.8	40.6/81.7	64.5	89.9	4.9/**65.7**	49.2	47.7	61.2/45.9
+EmDg 3B	48.6	73.9/43.8	15.0/43.7	38.0/77.7	62.9	88.6	4.7/62.7	55.7	35.2	53.6/42.7
+EmDg 11B	66.4	85.3/46.3	15.1/44.7	40.9/84.1	65.0	89.9	**5.3**/65.5	**56.6**	37.0	61.6/45.8
+Exp 3B	47.4	74.6/44.0	14.2/43.5	37.9/80.9	60.9	86.5	4.9/62.3	55.2	71.8	54.8/43.4
+Exp 11B	65.0	85.6/46.5	14.9/44.7	40.7/84.6	64.5	89.8	4.8/65.5	56.5	**73.5**	63.6/46.3
+TwSt 3B	46.6	78.0/44.5	14.6/43.7	37.3/77.5	60.4	86.8	5.2/61.9	55.3	72.4	74.8/56.5
+TwSt 11B	64.4	**85.9/46.6**	14.6/44.7	40.7/85.5	**65.8**	89.8	4.8/65.2	56.2	73.0	**74.4/57.9**
rev_final	48.8	83.3/45.4	14.6/43.9	39.0/81.6	61.2	88.6	4.4/61.9	55.0	72.4	73.2/57.3

Table 7: Progressive results for T0 3B and 11B results for continual training set up with best 3B results underlined & best 11B results bolded. T0zs denotes T0 zero-shot and is the average accuracy obtained on 12 eval datasets. B4, R1, BS denote BLEU-4, ROUGE-1 and BERTScore.