Impossible Distillation: from Low-Quality Model to High-Quality Dataset & Model for Summarization and Paraphrasing

Jaehun Jung[†] **Peter West**[†‡] **Liwei Jiang**[†‡] **Faeze Brahman**[†‡]
Ximing Lu[†‡] **Jillian Fisher**[†] **Taylor Sorensen**[†] **Yejin Choi**[†‡]
[†]Paul G. Allen School of Computer Science & Engineering, University of Washington
[‡]Allen Institute for Artificial Intelligence
hoony123@cs.washington.edu

Abstract

It is commonly perceived that the strongest language models (LMs) rely on a combination of massive scale, instruction data, and human feedback to perform specialized tasks – *e.g.* summarization and paraphrasing, without supervision. In this paper, we propose that language models can learn to summarize and paraphrase sentences, with none of these 3 factors. We present IMPOSSIBLE DISTILLATION, a framework that distills a task-specific dataset directly from an off-the-shelf LM, even when it is impossible for the LM itself to reliably solve the task. By training a student model on the generated dataset and amplifying its capability through self-distillation, our method yields a *high-quality* model and dataset from a *low-quality* teacher model, without the need for scale or supervision. Using IMPOSSIBLE DISTILLATION, we are able to distill an order of magnitude smaller model (with only 770M parameters) that outperforms 175B parameter GPT-3, in both quality and controllability, as confirmed by automatic and human evaluations. Furthermore, as a useful byproduct of our approach, we obtain DIMSUM+, a high-quality dataset with 3.4M sentence summaries and paraphrases. Our analyses show that this dataset, as a purely LM-generated corpus, is more diverse and more effective for generalization to unseen domains than all human-authored datasets – including Gigaword with 4M samples.

1 Introduction

The success of large language models (LLMs) has led to a paradigm shift in NLP research—tasks such as sentence summarization and paraphrasing can now be done without task-specific supervision, simply by prompting LLMs with instructions [27, 51, 50]. The stellar performance of LLMs, however, comes with costs: training LLMs to solve unsupervised tasks often requires multi-billion scale models, instruction data, and human feedback [32, 22, 14]. A natural question arises in this paradigm: does the task-solving capability uniquely emerge in the massive-scale, instruction-following LMs? If smaller, off-the-shelf LMs (*e.g.* GPT-2) do possess latent knowledge for these tasks, can we make use of this knowledge to train an efficient, yet powerful task model?

We present IMPOSSIBLE DISTILLATION, a novel distillation framework allowing off-the-shelf LMs to perform specialized tasks – sentence summarization and paraphrasing – without the need for scale or supervision. Our framework operates by (1) directly generating a task-specific dataset from an off-the-shelf LM, then (2) distilling a model using the dataset, thereby requiring neither a massive scale model nor curated human supervision. Aside from its applicability, IMPOSSIBLE DISTILLATION

Preprint. Under review.

Paraphrase (News Domain)	**Paraphrase (Biomedical Domain)**
Sentence *x*: At issue is a change in work rules that the company says will help reduce a massive surplus of processed steel.	**Sentence *x***: It is likely that the evidence from other settings, such as those in which birth size was decided by fetal ultrasound, will yield similar estimates.
Paraphrase *y*: The dispute is over a proposed change to the company's working conditions that the company says will help it reduce the amount of surplus steel.	**Paraphrase *y***: The findings should be expected to be generalizable to other settings, including those in which birth size is determined by fetal ultrasound.
Summary (Reddit Domain)	**Summary (Biomedical Domain)**
Sentence *x*: I've mentioned this to a few other people, and it seems that everyone else thinks this is completely weird, I don't know why.	**Sentence *x***: Additionally, the in vivo assays using P. berghei infected mice can be used as an alternative to screen more potent compounds for treating malaria.
Summary *y*: I've been telling people about it and they all think it's a weird thing to do.	**Summary *y***: The in vivo studies can be used as a platform to screen novel antimalarial compounds for use in malaria therapy.

Table 1: Samples in DIMSUM+. **All input-output pairs are generated by ∼1.6B LMs, without human supervision.** IMPOSSIBLE DISTILLATION distills a task-specific dataset and model from off-the-shelf LMs across domains, without scale or supervision. More examples in Appendix E.

is extremely powerful, enabling even small LMs (with <1B parameters) to outperform orders of magnitude larger LMs (*e.g.* GPT-3, with 175B parameters), all without task-specific supervision.

In IMPOSSIBLE DISTILLATION, dataset generation involves searching for high-quality input-output pairs (*e.g.* sentence-summary pairs) for the given task, using only an off-the-shelf LM (*e.g.* GPT-2), *i.e.* with no help of instruction-tuned models or initial data of any form. The key idea for making this process tractable is to (1) effectively reduce down the LM search space for input-output pairs through constrained decoding, and (2) ensure high-quality distillation with post-generation filters, derived from an explicit definition of the target task. By training a student model on this generated dataset, then further amplifying its capability through self-distillation, we yield a compact, yet strong end-stage model that outperforms much larger LMs in both automatic and human evaluation.

IMPOSSIBLE DISTILLATION is entirely independent of large and costly models or task-specific supervision, allowing us to distill the student model from any selection of initial LM (or a combination of LMs). In practice, we distill a compact task model (770M parameters) from 3 distinct LMs (all with ∼1.6B parameters), covering news / reddit / biomedical domains. Despite its size, the distilled model remarkably outputs more controllable, yet higher-quality summaries and paraphrases than 200 times larger GPT-3. Moreover, as a natural byproduct of this distillation, we obtain DIMSUM+, a large-scale sentence-level summarization and paraphrasing dataset with total of 3.4M pairs. Importantly, we find that DIMSUM+, although purely LM-generated, actually exhibits more lexical diversity and wider range of summary types than human-authored datasets. It even shows better adaptability to unseen domains: on an out-of-domain test set, a summarizer trained on our dataset outperforms the same model trained on the larger, human-authored Gigaword [59].

More broadly, our work shows that small, off-the-shelf LMs can simulate a rich source of task-specific knowledge, even when the model itself cannot reliably solve the task. By identifying and amplifying this knowledge into a high-quality dataset, IMPOSSIBLE DISTILLATION demonstrates a promising way of training task models through an efficient, effective, and reusable pipeline.

2 IMPOSSIBLE DISTILLATION

As shown in Figure 1, IMPOSSIBLE DISTILLATION starts from an off-the-shelf LM[1], then distills its task-specific knowledge based on a two-stage process of *decoding-guided distillation* and *self-distillation*. Our framework does not involve extra resource of human-written sentences, and specifically requires two inputs: a teacher model $\mathcal{M}_{\text{LM}}$ and a student model $\mathcal{M}_0$, which can all be initialized from generative LMs.

[1]While our method supports distilling from multiple initial LMs, we explain with a single LM for clarity.

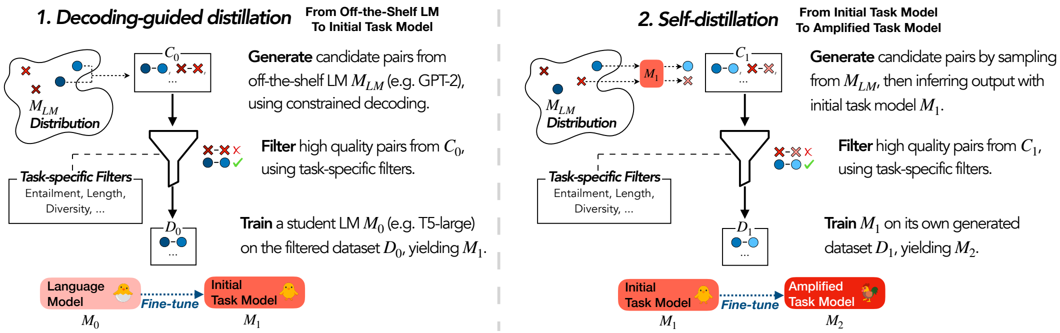

Figure 1: Overview of IMPOSSIBLE DISTILLATION. Starting from a small, off-the-shelf LM, we gradually produce higher-quality dataset and task model, **outperforming even the 200 times larger GPT-3 in both summarization and paraphrasing.**

In *decoding-guided distillation*, our goal is to directly generate a task-specific dataset $\mathcal{D}_0$ from scratch, using only the pre-trained teacher model $\mathcal{M}_{\text{LM}}$. To generate a high-quality dataset with minimal intervention to the model, we leverage an *overgenerate-filter* strategy: first generate a large pool of input-output pairs using $\mathcal{M}_{\text{LM}}$, then leave only the ones that qualify for the target task (*e.g.* meaningful sentence-summary pairs) using post-generation filters. Then, we use $\mathcal{D}_0$ to fine-tune $\mathcal{M}_0$ into an initial task model ($\mathcal{M}_0 \to \mathcal{M}_1$). In *self-distillation*, the initial task model $\mathcal{M}_1$ is further refined by fine-tuning on its own high-quality generations. We generate candidate pairs using $\mathcal{M}_{\text{LM}}$ and $\mathcal{M}_1$, filter high-quality pairs into $\mathcal{D}_1$, then train $\mathcal{M}_1$ on this dataset to amplify its capability ($\mathcal{M}_1 \to \mathcal{M}_2$).

By iterating over a *generate-filter-train* loop across the two stages, we gradually distill a higher quality dataset ($\mathcal{D}_0 \to \mathcal{D}_1$) and a stronger task model ($\mathcal{M}_0 \to \mathcal{M}_1 \to \mathcal{M}_2$). In the rest of this section, we illustrate the specifics of each stage (§2.1 – §2.2), and how we use the pipeline to execute the overall distillation (§2.3).

2.1 Decoding-guided Distillation Stage

2.1.1 Generating candidate pairs

Given our task of interest, we first generate a large pool of candidate input-output pairs $\mathcal{C}_0 = \{(x_1, y_1), \cdots, (x_{|\mathcal{C}_0|}, y_{|\mathcal{C}_0|})\}$ from an off-the-shelf LM $\mathcal{M}_{\text{LM}}$. The key challenge here lies in the low sample-efficiency of generated pairs. For example, a naive way of pair generation – just sampling x and y independently from $\mathcal{M}_{\text{LM}}$ – will not result in any meaningful pair that passes the task-specific filters. Prior works compensate for this low sample-efficiency by prompting LLM with task instructions [70, 61], but our method do not assume $\mathcal{M}_{\text{LM}}$ is few-shot promptable or instruction-following. This motivates us to impose a set of constraints as a strong prior for the LM decoding algorithm, which can be adopted by any $\mathcal{M}_{\text{LM}}$ and effectively reduces down the search space for candidate pairs. By simply imposing these constraints, we surprisingly find a large population of valid task pairs.

Contextual Constraints We begin by imposing *contextual constraints*, by first sampling a left context c_i from $\mathcal{M}_{\text{LM}}$, then conditioning the generation of both x_i and y_i on c_i. Intuitively, this constrains both sides of each pair to be a natural completion of the shared context, increasing the pairwise semantic coherence without resorting to an external source of context (*e.g.* human-written sentences). As shown in Figure 2, we collect c_i by generating 1-5 sentences from $\mathcal{M}_{\text{LM}}$ given a simple domain prefix. More details on contextual constraints are provided in Appendix A.1.

Sequential Generation with Lexical Constraints Inspired by an empirical observation that good summaries and paraphrases tend to preserve salient keywords in the original sentence, we consider the *sequential generation* of (x_i, y_i) with lexical constraints. As shown in Figure 2, we first generate x_i given c_i as the prefix, then also generate y_i given c_i but additionally constrained to include the keywords in x_i, extracted using an off-the-shelf keyword extraction tool [24]. Specifically, we employ Neurologic [44], a constrained decoding algorithm based on beam search to generate top k_1 candidate y_is per each x_i:

$$x_i \sim P_{\mathcal{M}_0}(\cdot|c_i); \quad \{y_{i1}, \cdots y_{ik_1}\} = \texttt{Neurologic}_{\mathcal{M}_{\text{LM}}}(\cdot|c_i; \texttt{keyword}(x_i)) \tag{1}$$

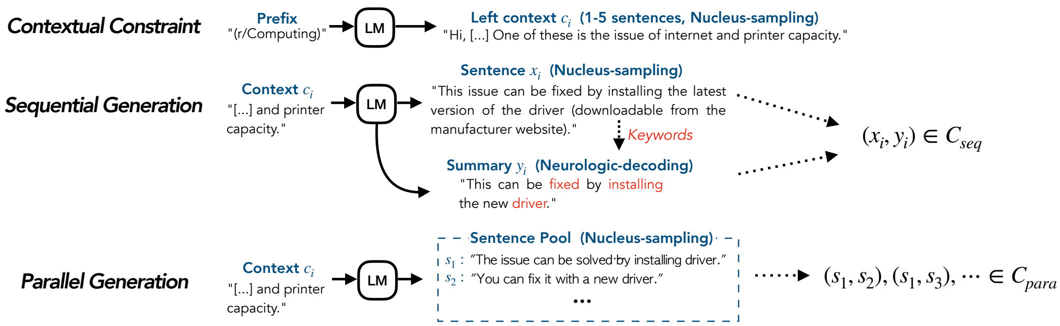

Figure 2: By imposing constraints in the decoding process of off-the-shelf LMs, we effectively reduce down the search space for task-specific pair generation. More examples shown in Appendix E.

For each c_i, this process yields k_1 candidate pairs: $\mathcal{C}_{\text{seq},i} = \{(x_i, y_{i1}), \cdots, (x_i, y_{ik_1})\}$. Aggregating pairs across multiple c_is, we obtain $\mathcal{C}_{\text{seq}} = \bigcup_i \mathcal{C}_{\text{seq},i}$.

Parallel Generation with Sampled Sentences While sequential generation preserves the salient spans of x in their surface-form, we also note that important phrases are often abstracted to shorter expression in good summaries and paraphrases. Hence, as an alternative to the extractive sequential generation, we introduce the *parallel generation* of pairs with stochastic decoding. We first sample a pool of k_2 sentences given c_i as prefix from $\mathcal{M}_{\text{LM}}$ using Nucleus-Sampling [29], then enumerate candidate pairs as the combination of these sentences:

$$\{s_{i1}, \cdots, s_{ik_2}\} = \texttt{Nucleus-Sampling}_{\mathcal{M}_{\text{LM}}}(\cdot|c_i; \tau_p) \tag{2}$$

$$\mathcal{C}_{\text{para},i} = \{(s_{im}, s_{in})|m, n \in [1, k_2], m \neq n\} \tag{3}$$

Here, τ_p is the top-p threshold. Note that this process does not impose any surface-level constraint to the generated sentences; we find that lowering the top-p threshold ($\tau_p = 0.7$) and hence sampling from a narrower subset of vocabulary suffices to induce a sample-efficient set of candidate pairs (Appendix D). Collecting the pairs across multiple c_is, we obtain $\mathcal{C}_{\text{para}} = \bigcup_i \mathcal{C}_{\text{para},i}$.

Finally, we define the initial candidate set as the union of two sets of generated pairs: $\mathcal{C}_0 = \mathcal{C}_{\text{seq}} \cup \mathcal{C}_{\text{para}}$. Broadly seen, the sequential generation yields a high-precision, extractive set of pairs, while the parallel generation results in a diverse, abstractive set of pairs. The heterogeneous properties of the two process enrich the sample diversity of our generated dataset.

2.1.2 Filtering for the high-quality pairs

Next, we filter the subset of candidate pairs $\mathcal{D}_0$ that qualify as good task-specific examples. Below, we first elaborate each of the filters with sentence summarization as the target task, then discuss how it generalizes to paraphrase generation.

Entailment Filter A faithful summary should be logically entailed by the original statement without hallucinating unsupported content. NLI models are well-suited to quantify this relationship, as they are trained to detect the logical entailment between an arbitrary pair of statements [10, 38]. Hence, we define a binary filter based on a small NLI model [41], and discard the pairs that do not achieve the entailment score over a predefined threshold τ_{entail}:

$$f_{entail}(x, y) = \mathbb{1}\Big\{P_{NLI}(x \Rightarrow y) \geq \tau_{entail}\Big\} \tag{4}$$

Length Filter A good summary should be a concise representation of the original statement. We therefore discard all pairs whose compression ratio (i.e. the sequence length ratio of y to x) is larger than a predefined threshold τ_{comp_ratio}:

$$f_{comp\_ratio}(x, y) = \mathbb{1}\Big\{|y| < |x| \cdot \tau_{comp\_ratio}\Big\} \tag{5}$$

Diversity Filter Our generation process decodes a large pool of pairs from a shared prefix c, which often results in multiple pairs having similar x or y. To remove such duplicate pairs, we employ a diversity filter $f_{diversity}$. Concretely, we define two pairs (x_1, y_1) and (x_2, y_2) to be duplicate when one pair entails another, either on the input side ($x_1 \Rightarrow x_2$) or the output side ($y_1 \Rightarrow y_2$). The diversity filter operates by first grouping all entailing pairs, then discarding all but one with the largest entailment score $P_{NLI}(x \Rightarrow y)$. In practice, this filter can be efficiently implemented using graph traversal; we detail the formal algorithm in Appendix A.2.

Incorporating all filters, we filter the task-specific dataset $\mathcal{D}_0$ as following:

$$\mathcal{D}_0 = \{(x, y)|(x, y) \in \mathcal{C}_0, f_{entail} \wedge f_{comp\_ratio} \wedge f_{diversity}(x, y) = 1\} \tag{6}$$

Generalizing to Paraphrase Our distillation process is grounded on the explicit definition of the target task, which allows the framework to generalize to paraphrase generation by simply redefining the filters. In general, a good paraphrase y should bear a bidirectional entailment with the original x, while not being too short or long compared to x. These assumptions are reflected in the corresponding updates to the respective filters:

$$f_{entail}(x, y) = \mathbb{1}\Big\{\min\big(P_{NLI}(x \Rightarrow y), P_{NLI}(y \Rightarrow x)\big) \geq \tau_{entail}\Big\} \tag{7}$$

$$f_{comp\_ratio}(x, y) = \mathbb{1}\Big\{|x| \cdot \tau_{comp\_ratio,1} \leq |y| < |x| \cdot \tau_{comp\_ratio,,2}\Big\} \tag{8}$$

Finally, an important property of a paraphrase is that it should not be similar to the original statement on the surface level. Following prior works, we quantify this constraint using the two metrics – Density [25] and ROUGE-L [40] – that measure surface-form similarity of two statements:

$$f_{abstract} = \mathbb{1}\Big\{\max\big(\text{Density}(x, y), \text{ROUGE-L}(x, y)\big) \leq \tau_{abstract}\Big\} \tag{9}$$

Training Initial Task Model We finish the decoding-guided distillation stage by training an initial task model using the generated dataset $\mathcal{D}_0$. The student model $\mathcal{M}_0$ is fine-tuned into $\mathcal{M}_1$ by maximizing $\mathbb{E}_{(x,y)\sim\mathcal{D}_0}[\log P_{\mathcal{M}_1}(y|x)]$, *i.e.* the conditional log-likelihood of y given x.

2.2 Self-Distillation Stage

Next, the task capability of $\mathcal{M}_1$ is further amplified into $\mathcal{M}_2$ through self-distillation. To generate candidate pairs without using human-written sentence data, we sample the input sentence x directly from teacher LM $\mathcal{M}_{\text{LM}}$, then generate the output sentence y by feeding x into the task model $\mathcal{M}_1$:

$$\mathcal{C}_1 = \big\{(x_1, y_1), \cdots | x_i \sim P_{\mathcal{M}_{\text{LM}}}(\cdot);\ y_i \sim P_{\mathcal{M}_1}(\cdot|x_i)\big\} \tag{10}$$

Using the same filters as the previous stage, we filter the high-quality pairs into $\mathcal{D}_1$. Finally, we fine-tune $\mathcal{M}_1$ on $\mathcal{D}_1$, yielding the end-stage model $\mathcal{M}_2$. Consistent with the prior findings on self-distillation [55, 2], this simple process significantly improves the performance of our task model (§3.4). In addition, our self-distillation outputs a large-scale, standalone dataset that can be evaluated and reused, *e.g.* to directly train a task model without re-iterating the distillation procedure (§3.3).

2.3 Distillation pipeline

In this section, we detail the distillation pipeline we apply in IMPOSSIBLE DISTILLATION. We start from 3 off-the-shelf LMs, and distill a single, powerful model $\text{T5}_{\text{IMPDISTILL}}$ capable of both (1) controllable sentence summarization and (2) paraphrasing across multiple domains.

Initial dataset We first generate the initial dataset $\mathcal{D}_0$ from off-the-shelf LMs. Our goal here is to synthesize a large-scale, multi-domain dataset for both summarization and paraphrasing. To do this, we start off 3 pre-trained LMs, GPT-2 [56], CTRL [35], BioGPT [45] – all with ∼1.6B parameters – generating pairs in news, reddit, biomedical domain respectively. We first sample 150k samples of c_is, then generate candidate pairs with each c_i as the left context. Filtering these pairs with the respective set of filters for summarization and paraphrasing, we yield $\mathcal{D}_0$ with 380k pairs (220k for summarization and 160k for paraphrasing).

Quantizing D_0 for Controllability While a student model can be trained directly on the initial dataset, prior works show that such a model typically lacks control over the important properties of generated sequences (e.g. summary length), resulting in sub-optimal performance [18]. Through IMPOSSIBLE DISTILLATION, endowing controllability to the student model is straightforward: we quantize the dataset based on controlled properties, then simply train the model with a control code

Dataset	Turk				$\mathbf{QQP_{summ}}$				$\mathbf{QQP_{para}}$		
Model	*R-1*	*R-2*	*R-L*	*B-F1*	*R-1*	*R-2*	*R-L*	*B-F1*	*Self-BLEU*	*iBLEU*	*B-F1*
PEGASUS	**90.9**	**86.7**	**90.7**	96.2	68.1	51.4	66.3	95.3	43.7	7.32	**98.6**
Flan-T5	91.0	86.3	90.4	**96.4**	**69.2**	**52.5**	**67.5**	**95.5**	**42.5**	**7.39**	96.1
GPT-$3_{\text{few-shot}}$	70.1	48.7	63.5	93.4	62.2	38.5	56.9	94.3	29.8	5.53	96.6
GPT-$3_{\text{zero-shot}}$	67.7	44.8	59.6	91.7	59.2	34.8	55.2	94.6	**12.6**	5.38	94.8
Flan-$\text{T5}_{\text{few-shot}}$	46.5	33.7	44.8	85.7	54.7	35.3	52.5	93.1	81.8	5.60	**98.8**
Referee	66.2	42.4	59.1	89.2	59.3	34.2	54.6	94.1	-	-	-
$\text{T5}_{\text{IMPDISTILL}}$	**71.6**	**57.8**	**69.4**	**93.8**	**65.2**	**45.9**	**63.2**	**94.9**	36.2	**8.31**	96.0

Table 2: Automatic evaluation of $\text{T5}_{\text{IMPDISTILL}}$ and baseline methods on three benchmark datasets. $\text{T5}_{\text{IMPDISTILL}}$ outperforms all unsupervised baselines across all benchmarks, including 200x larger GPT-3 with few-shot examples. We differentiate supervised methods (Top 2 rows) from unsupervised methods, and mark the best performance in each group in bold. Following prior works, we report ROUGE-1/2/L and BERTScore F1 [78] for summarization, Self-BLEU, iBLEU *(α=0.8)* [62] and BERTScore F1 for paraphrase generation.

[35] for each group. In this work, we focus on the control over two aspects of summaries – length and abstractiveness, and quantize $\mathcal{D}_0$ into 5 groups of samples: {long / short}-{abstractive / extractive} summaries, and paraphrases. The specific criteria of quantization are described in Appendix A.3.

Training Multi-task Model We fine-tune T5-large [57] with 770M parameters on the quantized $\mathcal{D}_0$, yielding initial model $\mathcal{M}_1$. For each group, we prepend the given instruction to the input x (*e.g.* `Generate a long, abstractive summary of ...`) as control code, then train the model to maximize likelihood of output y. Next, we generate $\mathcal{D}_1$ by first sampling 2M input sentence x from $\mathcal{M}_{\text{LM}}$, then generating the 5 types of y per each x with $\mathcal{M}_1$. Filtering yields $\mathcal{D}_1$ consisting of 3.4M pairs (2.1M for summarization and 1.3M for paraphrasing), which we name *Dataset of impossibly distilled summaries + paraphrases*, or DIMSUM+. Finally, we fine-tune $\mathcal{M}_1$ with the newly generated $\mathcal{D}_1$, yielding the amplified task model $\mathcal{M}_2$. We call this end-stage model $\text{T5}_{\text{IMPDISTILL}}$.

3 Experiments

Datasets We note that most pre-existing benchmarks for sentence summarization focus on news [59, 53, 52], which may not represent model performance across domains. To evaluate $\text{T5}_{\text{IMPDISTILL}}$ across news, reddit and biomedical domain, we collect 300 sentences from human-written corpora in each domain – XSUM [47], TL;DR [66], PubMed [48] – and compare the model summaries through human evaluation (for supervised baselines, we use Gigaword [59] as the train set). For automatic evaluation, we use existing benchmarks: Turk [72] and QQP [12]. Turk is a test-only benchmark, hence we follow prior works [21, 3] to use WikiAuto [33] as the train set for supervised baselines. QQP is originally designed for duplicate question detection, thus we filter only the duplicate question pairs, and segregate them for summarization and paraphrasing based on the compression ratio (< 0.8 for summarization, paraphrasing otherwise). We name these subsets QQP_{summ} and QQP_{para}.

Baselines We compare $\text{T5}_{\text{IMPDISTILL}}$ with both the unsupervised and supervised baselines. For unsupervised baselines, we include GPT-3 (`text-davinci-003`) in 5-shot and zero-shot setting, 5-shot Flan-T5-large, and Referee [61], an unsupervised summarizer distilled from GPT-3. For supervised baselines, we use PEGASUS-large [76] and Flan-T5-large [14] fine-tuned on each dataset.

Configuration Details We compare our end-stage model $\text{T5}_{\text{IMPDISTILL}}$ with baselines, unless otherwise specified. For dataset evaluation, we use DIMSUM, a summarization subset of DIMSUM+, and compare it with human-authored datasets for summarization. Except for the controllability experiments, we fix the control codes for $\text{T5}_{\text{IMPDISTILL}}$ to generate *long* and *abstractive* summaries. Additional implementation details including specific values of generation parameters and filter thresholds are provided in Appendix A.

3.1 Automatic Evaluation

Reference-based Evaluation In Table 2, we perform automatic, reference-based evaluation of $\text{T5}_{\text{IMPDISTILL}}$ and the baselines. In summarization (Turk, QQP_{summ}), $\text{T5}_{\text{IMPDISTILL}}$ significantly im-

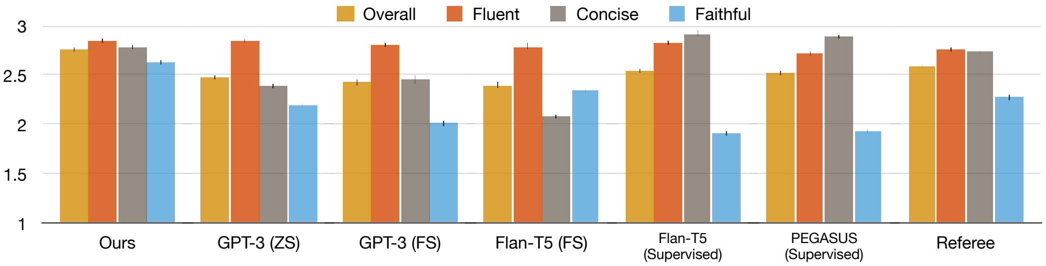

Figure 3: Human evaluation result of IMPOSSIBLE DISTILLATION and baselines (Krippendorf's alpha [31] = 0.61; substantial inter-annotator agreement), using 3-point Likert scale. $\text{T5}_{\text{IMPDISTILL}}$ is consistently preferred to the baselines, even including supervised models trained on Gigaword.

proves over all unsupervised methods across all metrics. Notably, $\text{T5}_{\text{IMPDISTILL}}$ outperforms both few-shot and zero-shot GPT-3. Moreover, $\text{T5}_{\text{IMPDISTILL}}$ is the only unsupervised model that marks higher iBLEU than the supervised baselines in paraphrasing (QQP_{para}). These results imply that the task performance does not come from the scale of the model alone, and a precise distillation algorithm can elicit stronger task performance from smaller LMs.

Controllability Evaluation Aside to its strong benchmark performance, our model supports control over the summary length and abstractiveness based on instruction. In Appendix C, we directly compare this controllability against instruction-following LMs, by few-shot prompting GPT-3 and Flan-T5 to generate summaries of specific types *({long / short}-{abstractive / extractive})*.

We find that instruction-following models cannot reliably follow the control instructions, even when they are specifically given the few-shot demonstrations that abide by the control code. For example, the mean compression ratio of "short" summaries generated by GPT-3 was 0.88, even though it was given 5 examples of short summaries (with compression ratio < 0.5). This is consistent to the previous findings that although GPT-3 generated summaries are controllable on a shallow level (*e.g.* for number of sentences in a paragraph summary [23]), they often violate constraints on a fine-grained level (*e.g.* for the total number of words in the summary [79]). In contrast, the short summaries from our model marked mean compression ratio of 0.479, demonstrating the effectiveness of our method for controllable summarization.

3.2 Human Evaluation

Reference-free Evaluation While reference-based metrics have been widely adopted in summarization domain [17], they may not correlate well with the human judgment of quality [43, 11, 60]. To compensate for the limitations of automatic evaluation, we directly assess the fluency, faithfulness, and conciseness of generated summaries through human evaluation (Figure 3). Consistent with the automatic evaluation, $\text{T5}_{\text{IMPDISTILL}}$ shows superior performance in all three dimensions compared to the baselines. We note that while the two supervised models and Referee exhibit high conciseness in their generations, their performance gain generally comes at the cost of faithfulness. On the contrary, $\text{T5}_{\text{IMPDISTILL}}$ generates fluent and concise summaries while staying faithful to the original statement, achieving higher overall score than all baselines. We present qualitative examples in Appendix F.

Domain	Ours vs. $\mathcal{M}_{\text{human}}$	Ours vs. $\mathcal{M}_{\text{mix}}$
News	**40.0** / 25.7 / 34.3	**39.3** / 23.0 / 37.7
Reddit	**37.0** / 35.0 / 28.0	31.7 / 32.3 / **36.0**
Bio	**38.7** / 26.0 / 34.3	**39.7** / 28.0 / 32.3

Table 3: Pairwise human evaluation on LM vs. human-written sentences for IMPOSSIBLE DISTILLATION. We report win / tie / lose ratio for each comparison.

LM-generated Sentences vs. Human-written Sentences Unlike prior works, IMPOSSIBLE DISTILLATION distills a task-specific dataset by generating both sides of input-output pairs. To analyze the effect of this purely LM-based distillation, we test an alternative way of generating dataset – by sampling human-written sentences from existing corpora (XSUM, TL;DR and PubMed), then summarizing them with $\mathcal{M}_1$ to produce $\mathcal{D}_1$. While fixing the dataset size, we generate two variants of $\mathcal{D}_1$: (1) $\mathcal{D}_{\text{human}}$, generated using only the human-written sentences, and (2) $\mathcal{D}_{\text{mix}}$, generated using 50-50 mix of human-written and LM-generated sentences. In Table 3, we present the human evaluation result comparing our model against the two models trained with the alternative sources of sentences ($\mathcal{M}_{\text{human}}$, $\mathcal{M}_{\text{mix}}$).

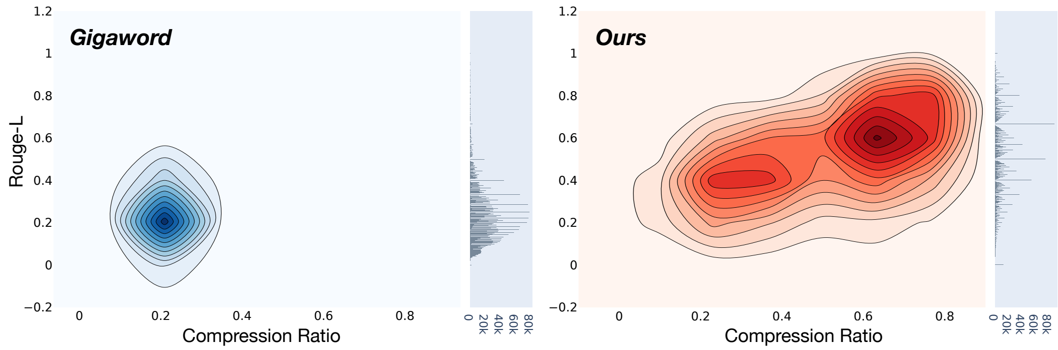

Figure 4: Distribution of summarization strategy in Gigaword (left), DIMSUM (right).

Dataset	H_1	H_2	H_3	MSTTR
Turk	9.92	14.25	15.11	0.302
QQP_{summ}	9.16	14.43	16.63	0.424
Gigaword	10.12	16.87	21.22	0.472
DIMSUM	**10.38**	**17.38**	**21.46**	**0.511**

Table 4: Lexical diversity of datasets. DIMSUM, while LM-generated, provides more lexical diversity than human-authored datasets.

Configuration	R-L	B-F1
In-domain supervision (100%)	67.5	95.5
Gigaword only	58.1	84.7
Gigaword + In-domain (100%)	60.5	89.6
DIMSUM only	62.1	94.2
DIMSUM + In-domain (50%)	68.3	95.8
DIMSUM + In-domain (100%)	**70.9**	**96.0**

Table 5: Performance of T5-large on QQP_{summ} with different training configurations.

We find here that $T5_{\text{IMPDISTILL}}$, purely trained on LM-generated sentences, are generally preferred than the models trained with human-written sentences. The results imply that merely random-sampling sentences from existing corpus may not suffice to create a high-quality dataset; generating sentences with the right choice of LM and decoding algorithm could be a promising alternative, as the LMs are pre-trained with an exact objective to represent the human text distribution.

3.3 Dataset Quality Evaluation

Next, we directly compare the quality of our generated dataset against conventional summarization datasets. We use 3 human-authored datasets: Gigaword, Turk and QQP_{summ} as baselines, and evaluate the diversity and usefulness of DIMSUM against them.

DIMSUM is more diverse than human-authored datasets. We explore the diversity of summarization samples in DIMSUM and baseline datasets. First, we compare the summarization strategy diversity, *i.e.* the diversity of pairs in terms of abstractiveness and compression ratio. In Figure 4 and Appendix B, we plot the summarization strategy distribution of the train split in each dataset, with ROUGE-L and compression ratio as the two axes. The plots clearly present the superior diversity of DIMSUM than the human-authored datasets. Notably, while Gigaword consists of 4M human-written summaries, its distribution is biased to a very specific region of abstractiveness and compression ratio. Our dataset, despite being smaller than Gigaword, presents a well distributed set of summaries across all region of ROUGE-L and compression ratio, providing rich supervision signal to the trained model.

In addition, we analyze the lexical diversity of each dataset in Table 4. Following [21], we gauge the 1/2/3-gram entropy and the mean segmented token type ratio (MSTTR) of sentences in each dataset. Again, our dataset provides the largest diversity in all metrics, powered by the extensive distillation across multiple domains.

DIMSUM better generalizes to unseen domain. To validate whether DIMSUM is helpful for generalizing to unseen domain, we directly train T5 on Gigaword and DIMSUM, then test it on QQP_{summ}. The results are shown in Table 5 (*Gigaword only,* DIMSUM *only*). Compared to the Gigaword-trained model, the model trained on DIMSUM performs much closer to the *In-domain supervision*, attesting to the generalizability of our dataset to unseen domain.

DIMSUM is effective for transfer learning. As shown in the diversity analysis, human-authored datasets typically cover a narrow, specialized style and domain [25]; in contrast, IMPOSSIBLE DIS-

TILLATION induces a large-scale, multi-domain dataset of sentence-summary pairs. This motivates us to consider another use-case of DIMSUM, where the synthetic examples are used to train a general summarizer, which can be fine-tuned to the specific style and domain of human-written benchmarks. We validate this scenario in Table 5, by first fine-tuning T5 on either Gigaword or DIMSUM, then further training the model on the in-domain train set of QQP_{summ}.

While fine-tuning on Gigaword degrades the test set performance *(Gigaword + In-domain (100%))*, training on DIMSUM improves performance over purely in-domain supervised model *(*DIMSUM + *In-domain (100%))*. Moreover, a summarizer trained on our dataset surpasses in-domain supervision, fine-tuning on only half of the in-domain train set *(*DIMSUM + *In-domain (50%))*. This substantiates the usefulness of our data for transfer learning, from a general task model to a specialized task model.

3.4 Ablation Study

Does self-distillation matter? We ablate the self-distillation of IMPOSSIBLE DISTILLATION in two ways. First, we omit the self-distillation stage and test the *initial model* $\mathcal{M}_1$. In this case, ROUGE-L on Turk degrades by 10% relatively to $\text{T5}_{\text{IMPDISTILL}}$, indicating the importance of self-distillation in amplifying the model capability. Next, we consider directly fine-tuning off-the-shelf T5 on DIMSUM+, rather than distilling $\mathcal{M}_1$ on this dataset. Although the high-quality samples in DIMSUM+ drive competitive performance in this *directly-supervised* model, it stills falls behind the full $\text{T5}_{\text{IMPDISTILL}}$ performance, demonstrating the effectiveness of distilling further the initial task model.

Configuration	R-L	B-F1
Initial model $\mathcal{M}_1$	63.3	89.1
Direct supervision on $\mathcal{D}_1$	69.0	93.1
No control	68.5	93.3
Summarization only	69.1	**94.0**
$\text{T5}_{\text{IMPDISTILL}}$	**69.4**	93.8

Table 6: Ablation study on Turk dataset.

Does controllability matter? We also consider our method with *no control*, *i.e.* removing controllability from the end-stage model. Consistent with the prior findings [18], training on the quantized dataset yields slightly better performance than without quantization, even if we fix the control code during test time (*long-abstractive*).

Can we just train a task-specific model? Finally, we remove the paraphrase generation from the distillation pipeline and train a summarization-specific model. This *Summarization only* model performs comparable to $\text{T5}_{\text{IMPDISTILL}}$, which is capable of both summarization and paraphrasing. The result shows that while it is possible to train a model for a single specific task, training on multiple related tasks does not hurt the performance, attesting to the applicability of IMPOSSIBLE DISTILLATION on multi-task distillation.

4 Related Work

Unsupervised Summarization / Paraphrasing Conventional approaches for unsupervised summarization and paraphrasing have focused on task-specific surrogates – e.g. reconstruction of the original text [4, 8, 80, 58] – to supervise the model toward desired output. These surrogate tasks inherently provide a weak and sparse supervision signal compared to the complexity that the target tasks involve, often mandating a carefully engineered train loop [37] and auxiliary loss [4, 68]. Apart from the task-specific methods, a growing line of research seeks to harness LMs to summarize and paraphrase without supervision [13, 6, 19, 73]. Notably, recent findings suggest that zero-shot summaries prompted from LLMs exhibit higher quality than supervised models [23, 79].

Task-solving with Language Model More broadly, task-solving capabilities of LMs have been tested and analyzed across domains [27]. While large-scale pre-training allows models to acquire sufficient knowledge to solve complex tasks [7, 77, 49, 34], recent works suggest that their full capability is elicited from aligning the model knowledge with additional fine-tuning – e.g. using instruction data [14, 51, 69] and human feedback [81, 46] – which often requires a curated set of annotated data. In a sense, our work shows a promising alternative to this paradigm, by amplifying model capability based on the explicit definition of the target task, rather than human annotation.

Data Generation with Language Model Another line of related works propose to directly train models with LM-generated data, improving model reasoning [75, 28, 30], robustness [9], controllability [61] and language understanding [74, 20, 26]. These works essentially follow the conceptual framework of Symbolic Knowledge Distillation [70], where the teacher model's knowledge is transferred to a student model via a symbolic, textual dataset. Other works explore to extract a standalone corpus from LMs, spanning from knowledge base [5, 1], contextual dialogue [36], and model behavior evaluation [54]. However, these works typically impose a strong assumption on the generator LM [63, 16, 67], and require manually constructed set of prompts [5]. Overcoming these limitations, IMPOSSIBLE DISTILLATION generalizes data generation into a multi-task, off-the-shelf setup, removing the dependence to the underlying model's capability for data generation. In effect, we show that small LMs can be harnessed to generate a high-quality, reusable dataset for multiple tasks at hand.

5 Conclusion

In this work, we propose IMPOSSIBLE DISTILLATION, a novel distillation framework that significantly improves LM capability by accurately searching and amplifying its task-specific knowledge. We empirically show that IMPOSSIBLE DISTILLATION can empower small LMs to outperform their gigantic counterparts in both generation quality and controllability, across domains and tasks, without supervision. Also, DIMSUM+, the natural byproduct of our method, presents higher diversity and usability than human-authored baselines. IMPOSSIBLE DISTILLATION shows a promising direction to rediscover the under-explored capabilities of off-the-shelf language models, without resorting to external resource or extra supervision.

As with any distillation technique, IMPOSSIBLE DISTILLATION carries potential risk of amplifying undesirable properties of language models. While we focus on conditional generation tasks where the output is closely bound to the input, the trained model could inherit the bias and toxicity of its teacher in a more open-ended setting. Nonetheless, IMPOSSIBLE DISTILLATION distills knowledge into a symbolic, textual dataset – which can be interpreted and evaluated, allowing users to intervene in the distillation process and selectively filter which knowledge to be amplified. The inherent transparency of IMPOSSIBLE DISTILLATION, when incorporated with recent techniques for automatic bias detection and reduction, could empower safer knowledge transfer between language models.

6 Acknowledgements

This work was funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) (funding reference number 401233309), DARPA MCS program through NIWC Pacific (N66001-19-2-4031), and the Allen Institute for AI. We also thank OpenAI for providing access to the GPT-3 API.

References

[1] Dimitrios Alivanistos, Selene Báez Santamaría, Michael Cochez, Jan-Christoph Kalo, Emile van Krieken, and Thiviyan Thanapalasingam. Prompting as probing: Using language models for knowledge base construction. *arXiv preprint arXiv:2208.11057*, 2022.

[2] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *CoRR*, abs/2012.09816, 2020.

[3] Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 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, July 2020. Association for Computational Linguistics.

[4] Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. SEQˆ3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 673–681, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[5] Chandra Bhagavatula, Jena D. Hwang, Doug Downey, Ronan Le Bras, Ximing Lu, Keisuke Sakaguchi, Swabha Swayamdipta, Peter West, and Yejin Choi. I2d2: Inductive knowledge distillation with neurologic and self-imitation, 2022.

[6] Adithya Bhaskar, Alexander R Fabbri, and Greg Durrett. Zero-shot opinion summarization with gpt-3. *arXiv preprint arXiv:2211.15914*, 2022.

[7] Jillian Bommarito, Michael Bommarito, Daniel Martin Katz, and Jessica Katz. Gpt as knowledge worker: A zero-shot evaluation of (ai)cpa capabilities, 2023.

[8] Arthur Bražinskas, Mirella Lapata, and Ivan Titov. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169, Online, July 2020. Association for Computational Linguistics.

[9] Howard Chen, Jacqueline He, Karthik Narasimhan, and Danqi Chen. Can rationalization improve robustness? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3792–3805, Seattle, United States, July 2022. Association for Computational Linguistics.

[10] Jifan Chen, Eunsol Choi, and Greg Durrett. Can NLI models verify QA systems' predictions? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3841–3854, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[11] Wang Chen, Piji Li, and Irwin King. A training-free and reference-free summarization evaluation metric via centrality-weighted relevance and self-referenced redundancy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 404–414, 2021.

[12] Zihang Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. Quora question pairs, 2017.

[13] Bharath Chintagunta, Namit Katariya, Xavier Amatriain, and Anitha Kannan. Medically aware gpt-3 as a data generator for medical dialogue summarization. In Ken Jung, Serena Yeung, Mark Sendak, Michael Sjoding, and Rajesh Ranganath, editors, *Proceedings of the 6th Machine Learning for Healthcare Conference*, volume 149 of *Proceedings of Machine Learning Research*, pages 354–372. PMLR, 06–07 Aug 2021.

[14] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

[15] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018.

[16] Bosheng Ding, Chengwei Qin, Linlin Liu, Lidong Bing, Shafiq Joty, and Boyang Li. Is gpt-3 a good data annotator?, 2022.

[17] Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 2021.

[18] Angela Fan, David Grangier, and Michael Auli. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[19] Xiyan Fu, Yating Zhang, Tianyi Wang, Xiaozhong Liu, Changlong Sun, and Zhenglu Yang. RepSum: Unsupervised dialogue summarization based on replacement strategy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6042–6051, Online, August 2021. Association for Computational Linguistics.

[20] Jiahui Gao, Renjie Pi, LIN Yong, Hang Xu, Jiacheng Ye, Zhiyong Wu, WEIZHONG ZHANG, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. Self-guided noise-free data generation for efficient zero-shot learning. In *International Conference on Learning Representations*, 2023.

[21] Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online, August 2021. Association for Computational Linguistics.

[22] Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. Improving alignment of dialogue agents via targeted human judgements, 2022.

[23] Tanya Goyal, Junyi Jessy Li, and Greg Durrett. News summarization and evaluation in the era of gpt-3, 2022.

[24] Maarten Grootendorst. Keybert: Minimal keyword extraction with bert., 2020.

[25] Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[26] Jesse Michael Han, Igor Babuschkin, Harrison Edwards, Arvind Neelakantan, Tao Xu, Stanislas Polu, Alex Ray, Pranav Shyam, Aditya Ramesh, Alec Radford, and Ilya Sutskever. Unsupervised neural machine translation with generative language models only, 2021.

[27] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021.

[28] Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers, 2022.

[29] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

[30] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes, 2023.

[31] John Hughes. krippendorffsalpha: An r package for measuring agreement using krippendorff's alpha coefficient, 2021.

[32] Yunjie Ji, Yong Deng, Yan Gong, Yiping Peng, Qiang Niu, Lei Zhang, Baochang Ma, and Xiangang Li. Exploring the impact of instruction data scaling on large language models: An empirical study on real-world use cases, 2023.

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

[34] Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. Maieutic prompting: Logically consistent reasoning with recursive explanations. *arXiv preprint arXiv:2205.11822*, 2022.

[35] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation, 2019.

[36] Hyunwoo Kim, Jack Hessel, Liwei Jiang, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Le Bras, Malihe Alikhani, Gunhee Kim, Maarten Sap, and Yejin Choi. Soda: Million-scale dialogue distillation with social commonsense contextualization, 2022.

[37] Philippe Laban, Andrew Hsi, John Canny, and Marti A. Hearst. The summary loop: Learning to write abstractive summaries without examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5135–5150, Online, July 2020. Association for Computational Linguistics.

[38] Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177, 2022.

[39] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining, 2019.

[40] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[41] Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[42] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[43] Yizhu Liu, Qi Jia, and Kenny Zhu. Reference-free summarization evaluation via semantic correlation and compression ratio. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2109–2115, Seattle, United States, July 2022. Association for Computational Linguistics.

[44] Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. NeuroLogic decoding: (un)supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4299, Online, June 2021. Association for Computational Linguistics.

[45] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6), sep 2022.

[46] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. Teaching language models to support answers with verified quotes, 2022.

[47] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[48] National Library of Medicine. Pubmed.

[49] Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Gpt3-to-plan: Extracting plans from text using gpt-3. *arXiv preprint arXiv:2106.07131*, 2021.

[50] OpenAI. Gpt-4 technical report, 2023.

[51] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[52] Paul Over and Walter Liggett. Introduction to duc-2004: An intrinsic evaluation of generic news text summarization systems. In *Proceedings of Document Understanding Conference*, 2004.

[53] Paul Over and James Yen. Introduction to duc-2003: An intrinsic evaluation of generic news text summarization systems. In *Proceedings of Document Understanding Conference*, 2003.

[54] Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Ben Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemí Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. Discovering language model behaviors with model-written evaluations, 2022.

[55] Minh Pham, Minsu Cho, Ameya Joshi, and Chinmay Hegde. Revisiting self-distillation, 2022.

[56] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *openai*, 2019.

[57] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

[58] Aurko Roy and David Grangier. Unsupervised paraphrasing without translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6033–6039, Florence, Italy, July 2019. Association for Computational Linguistics.

[59] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[60] Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3246–3256, Hong Kong, China, November 2019. Association for Computational Linguistics.

[61] Melanie Sclar, Peter West, Sachin Kumar, Yulia Tsvetkov, and Yejin Choi. Referee: Reference-free sentence summarization with sharper controllability through symbolic knowledge distillation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9649–9668, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[62] Hong Sun and Ming Zhou. Joint learning of a dual SMT system for paraphrase generation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 38–42, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[63] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[64] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[65] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models, 2018.

[66] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[67] Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. Want to reduce labeling cost? gpt-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, 2021.

[68] Yaushian Wang and Hung-Yi Lee. Learning to encode text as human-readable summaries using generative adversarial networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4187–4195, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[69] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks, 2022.

[70] Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625, Seattle, United States, July 2022. Association for Computational Linguistics.

[71] BigScience Workshop. Bloom: A 176b-parameter open-access multilingual language model, 2023.

[72] Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415, 2016.

[73] Xianjun Yang, Yan Li, Xinlu Zhang, Haifeng Chen, and Wei Cheng. Exploring the limits of chatgpt for query or aspect-based text summarization. *arXiv preprint arXiv:2302.08081*, 2023.

[74] Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

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

[76] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *ArXiv*, abs/1912.08777, 2019.

[77] Lining Zhang, Mengchen Wang, Liben Chen, and Wenxin Zhang. Probing GPT-3's linguistic knowledge on semantic tasks. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 297–304, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[78] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.

[79] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. Benchmarking large language models for news summarization, 2023.

[80] Jiawei Zhou and Alexander Rush. Simple unsupervised summarization by contextual matching. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5101–5106, Florence, Italy, July 2019. Association for Computational Linguistics.

[81] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020.

A Implementation Details

A.1 Generating pairs

IMPOSSIBLE DISTILLATION generates each candidate pair in 3 domains using an off-the-shelf LM for the respective domain: GPT-2 (news), CTRL (reddit), and BioGPT (biomedical). Here, we first generate 1-5 sentences from each LM as the contextual constraint c_i. For reddit and biomedical domain, this process is straightfoward as the two LMs are pre-trained to generate sentences in the corresponding domain: for CTRL, we use the predefined control codes for reddit-style generation (*e.g.* (`r/Gaming`)), and for BioGPT, we free-form generate without any prefix given. For CTRL control codes, we refer the readers to the original paper [35]. With GPT-2, we find that formatting a simple prefix including a city and a media name (*e.g.* `London`, (`CNN`) –) suffices to generate high-quality news-style sentences without domain adaptation.

For sequential pair generation, we use KeyBERT [24], an off-the-shelf keyword extracting library to extract at most 5 keywords from each sentence x, and generate $k_1 = 10$ summaries per x. For parallel pair generation, we set $k_2 = 100$ and $\tau_p = 0.7$. Since the decoding process leverages a shared prefix c_i to generate a large pool of candidate pairs, the computation is highly parallelizable, and we use 8 Quadro RTX 8000 GPUs to run all our experiments.

A.2 Filtering for high-quality pairs

For the entailment filter, we use RoBERTa-large [42] fine-tuned on WANLI [41] as the NLI model, and set $\tau_{entail} = 0.9$ to ensure only the pairs with strong entailment are filtered. For summarization, we set $\tau_{comp\_ratio} = 0.8$, such that the summary has at most 80% number of tokens compared to the original sentence. For paraphrasing, we constrain the length of y to be in the range of 80% $\sim$ 150% of the original length , *i.e.* $\tau_{comp\_ratio,1} = 0.8$ and $\tau_{comp\_ratio,2} = 1.5$. Also, we use $\tau_{abstract} = 0.6$ in the abstractiveness filter for paraphrase generation.

Finally, we present the formal algorithm of the diversity filter in Algorithm 1. We first create an undirected graph G where pairs are nodes and edges exist between duplicate pairs, then find the set S of all connected components in G. By discarding all but the one with the maximal entailment score in each component, we effectively remove the duplicate pairs in the candidate pool. As the duplicate pair search with NLI model is parallelizable, the time complexity follows that of the connected component search, *i.e.* $O(|P| + |E|)$ when using DFS-based algorithm [64].

Algorithm 1 Diversity Filter

Input: A set of pairs $\mathcal{P}_{\text{in}} = \{(x_1, y_1), \cdots, (x_{|P|}, y_{|P|})\}$ generated using the same prefix c
Output: Filtered set of pairs $\mathcal{P}_{\text{out}}$

```
E ← ∅
for i, j ∈ [1, |P|], i ≠ j do  // search for duplicate pairs
    if P_NLI(x_i ⇒ x_j) > τ_entail then
        E ← E ∪ {(x_i, y_i), (x_j, y_j)}
    else if P_NLI(y_i ⇒ y_j) > τ_entail then
        E ← E ∪ {(x_i, y_i), (x_j, y_j)}
    end if
end for
G ← (P_in, E)  // define a graph where nodes are pairs and edges connect duplicate pairs
S ← Connected-Components(G)
P_out ← ∅
for C ∈ S do  // find the max-entailing pair in each connected component
    p_out = argmax_{(x,y)∈C} P_NLI(x ⇒ y)
    P_out ← P_out ∪ {p_out}
end for
```

A.3 Quantizing dataset for controllability

Prior to training the task model in each stage, we quantize the generated dataset into 5 groups: {long / short}-{abstractive / extractive} summaries, and paraphrases. To represent each pair (x, y) in terms of length and abstractiveness, we first quantify the compression ratio and surface-form similarity between x and y:

$$comp(x, y) = \frac{|y|}{|x|}, \quad sim(x, y) = \max\big(\text{Density}(x, y), \text{ROUGE-L}(x, y)\big) \tag{11}$$

Then, we group each pair in the dataset to be one of the 5 groups below based on the two metrics.

- *Short-Abstract Summary* : $comp(x, y) < 0.5,\ sim(x, y) < 0.6$
- *Short-Extractive Summary*: $comp(x, y) < 0.5,\ sim(x, y) \geq 0.6$
- *Long-Abstract Summary*: $0.5 \leq comp(x, y) < 0.8,\ sim(x, y) < 0.6$
- *Long-Extractive Summary*: $0.5 \leq comp(x, y) < 0.8,\ sim(x, y) \geq 0.6$
- *Paraphrase*: $0.8 \leq comp(x, y) < 1.5,\ sim(x, y) < 0.6$

This way, we not only train a multi-task model capable of controllable summarization and paraphrasing, but also obtain a large-scale dataset covering diverse summarization strategy, as illustrated in Appendix B.

B Dataset Evaluation

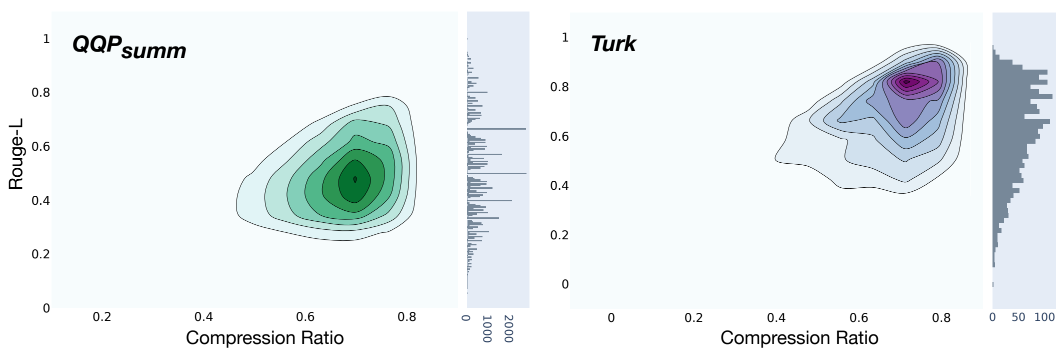

Figure 5: Distribution of summarization strategy in QQP_{summ} (left), Turk (right).

Dataset	*Short-Abstractive*	*Short-Extractive*	*Long-Abstractive*	*Long-Extractive*	*Paraphrase*	*Total*
Gigaword	3.54M	60k	168k	17k	18k	3.8M
QQP_{summ}	4.8k	1k	29.3k	15.2k	-	50.3k
QQP_{para}	-	-	-	-	68.7k	68.7k
DIMSUM+	574k	197k	711k	648k	1.33M	3.46M

Table 7: Number of examples for each pair type in the train split of $\mathcal{D}_1$, Gigaword, and QQP.

In Figure 5, we additionally plot the summarization strategy distribution of QQP_{summ} and Turk dataset. Since Turk does not provide the train split, we plot the distribution of the valid and test split of the dataset. Compared to DIMSUM+, these human-authored datasets exhibit relatively concentrated region of the summarization strategy space.

In Table 7, we also compare the number of examples for each pair type in the train split of DIMSUM+, Gigaword, and QQP. In Gigaword, majority of the examples represent short and abstractive summaries, as the dataset is constructed by collecting news headlines as proxies for sentence-level summaries. We also find that Gigaword includes 18k examples where the output is longer than 80% the length of the input, despite the dataset being a sentence summarization benchmark. Compared to the human-authored datasets, our dataset presents a large-scale, well-distributed set of pair types for summarization and paraphrasing.

C Controllability Evaluation

Attribute	Length (*Comp. Ratio*)			Abstractiveness (*ROUGE-L*)		
Model	*Long*	*Short*	Δ	*Extractive*	*Abstractive*	Δ
GPT-3$_{\text{few-shot}}$	1.07	0.88	0.19	56.2	54.0	2.2
Flan-T5$_{\text{few-shot}}$	0.62	0.29	0.33	65.1	52.3	12.8
T5$_{\text{IMPDISTILL}}$	0.72	0.48	0.23	77.1	51.3	25.8

Table 8: Experimental results on controllable summarization.

In this section, we compare the controllability of T5$_{\text{IMPDISTILL}}$ against few-shot prompted GPT-3 and Flan-T5 across summary length and abstractiveness. Using Turk dataset, we instruct each model to generate 4 types of summaries with instruction: "`Generate {long / short}, {abstractive / extractive} summary of the given sentence:`". To better guide the baseline models to the control instruction, we manually construct 5 few-shot examples for each summary group and append them to the instruction. We report the average compression ratio for long / short summaries and ROUGE-L of extractive / abstractive summaries from each model in Table 8.

Our model, explicitly trained with the quantized dataset, shows significantly more controllability then few-shot instructed LMs. Notably, GPT-3, when instructed to generate *long summary*, records mean compression ratio of 1.07 (*i.e.* generates longer summary then the original sentence on average). Flan-T5 shows better controllability over length, but still falls behind the abstractiveness control compared to our model. These results imply that while instructions and few-shot examples could signal some degree of control over the LM generations, they may not suffice to control more sparse and fine-grained properties of generations. IMPOSSIBLE DISTILLATION could be an effective alternative to these methods, as it allows control over any type of quantizable property, by generating a large pool of train samples and grouping them based on the desired property.

D Pair Generation Analysis

Generation Process	Sample Efficiency	Average ROUGE-L
Sequential Generation	0.32	75.5
Parallel Generation	1.15	58.6

Table 9: Sample efficiency and average ROUGE-L of generated pairs in sequential and parallel generation process.

In Table 9, we analyze the difference between the sequential generation and parallel generation in IMPOSSIBLE DISTILLATION. We first investigate the sample efficiency of each pair generation process, defined as the number of pairs that pass the summarization filters, divided by the number of contextual constraint c_is used to generate them.

From 150k c_is, sequential generation yields 48k sentence-summary pairs, marking the sample efficiency of 0.32. Meanwhile, parallel generation yields 172k pairs, hence the sample efficiency of 1.15. Note that in our experiment, we generate different number of candidate pairs from the two generation process, *i.e.* we used $k_1 = 10$ for sequential generation and $k_2 = 100$ for parallel generation. Therefore, the likelihood of a single pair passing the filter is actually higher in sequential generation than parallel generation. However, we empirically find that enlarging k_1 does not help in improving sample efficiency of sequential generation, as the beam-search based generations lacks diversity even with the larger beam size [65]. In contrast, parallel generation induces more than 1 pair per each c_i on average, thanks to the diversity of sentences enabled by stochastic decoding and larger sample size.

Next, we compare the average ROUGE-L between x and y in each generated pair. Sequential generation yields more extractive summaries than parallel generation, contributing to the overall coverage of summarization strategy in the generated dataset.

E Pair Generation Examples

Sequential Pair Generation (Summarization)	
Left Context c	There had been fears the flare could ignite the escaping gas at the Elgin platform, about 150 miles (240 km) east of the Scottish city of Aberdeen, potentially causing a huge explosion. Total said it had received the first indication that the flare might be out at lunchtime on Friday. The firm is “mobilizing all means to allow these options to be implemented," it said. The company, which is still investigating the cause of the leak, estimates that 200,000 cubic meters of gas a day are escaping.
Sentence x	“The gas cloud is fairly small in size and prevailing winds are blowing it away from the platform and dispersing it,” Total said.
Keywords $\texttt{keyword}(x)$	gas, cloud, small, blowing, Total
Summary y	The gas cloud is small and blowing away, Total said.

Sequential Pair Generation (Paraphrasing)	
Left Context c	The impact of obesity on health-related quality of life (HRQOL) in adolescents and young adults with spinal deformity is not well described.
Sentence x	The purpose of this study was to compare HRQOL measures in adolescent idiopathic scoliosis (AIS) patients with and without obesity.
Keywords $\texttt{keyword}(x)$	HRQQL, idiopathic, AIS, obesity
Paraphrase y	This study aimed to investigate the relationship between HRQOL and obesity in adolescents with idiopathic scoliosis (AIS).

Parallel Pair Generation (Summarization)	
Left Context c	A banana primarily consists of carbo hydrate chains (sugar), but also contains some minor amount of minerals and vitamins. Let’s see what happens with this stuff - Sugar: Will be broken down to either be stored as fat (another form of carbo hydrate chains) or broken up and used to provide cell energy; the resulting "waste" hydrogen and carbon is disposed of in form of CO2 or H2O. Minerals: Are mainly used to regenerate organs/tissue and other organ functions; these could probably be still in your body, but even if they are, they are probably very rare. Vitamins: The atoms are very often disposed after use, so they too leave your body. I am no expert, so i canť give you an answer as to whether they leave through urine or excrements.
Sentence x	They do leave in rather short time frames, because the body can’t store them well and needs it daily (that is why your diet should include them).
Summary y	They do leave in a short time, but the body does not store them long.

Parallel Pair Generation (Paraphrasing)	
Left Context c	As part of the process for the upcoming release of the Android M, Google is also adding a new camera API to the latest Android OS.
Sentence x	This API allows third-party apps to use the camera of Android devices.
Paraphrase y	The new API will allow developers to use Android’s camera features to create custom apps.

Table 10: Examples of constrained pair generation.

F Qualitative Examples

Sentence	Limitation of heat losses (due to the constriction of subcutaneous blood vessels) together with activation of the metabolism in muscles and other tissues provide for significant heat accumulation and the increase in body temperature of arousing ground squirrels.
PEGASUS (supervised)	heat loss in squirrels linked to constriction of blood vessels
T5 (supervised)	study finds squirrel metabolism may be linked to heat loss
GPT-3 (few-shot)	Studies found that body temperature, heat losses during arousal, due to the constriction of subcutaneous blood vessels.
GPT-3 (zero-shot)	The sentence discusses how the limitation of heat losses, along with the activation of metabolism, causes significant heat accumulation and an increase in body temperature of arousing ground squirrels.
$T5_{\text{ImpDistill}}$	The heat accumulation and the increase in body temperature of arousing ground squirrels are due to the reduced heat loss from the subcutaneous blood vessels and the increased metabolism of the muscles.
Sentence	In the case of life-sentence prisoners, the victim can obtain information about the minimum number of years the life prisoner must serve, when the life prisoner is being considered for release and have the opportunity to give views when the prisoner is considered for final release.
PEGASUS (supervised)	victims of life-sentence prisoners to have more rights
T5 (supervised)	victim can obtain information about life-sentence prisoner
GPT-3 (few-shot)	Victims can receive information about a prisoner's release, conditions, and breaches of those conditions, as well as information about life-sentence prisoners and their potential release.
GPT-3 (zero-shot)	The Victim Information Scheme allows victims to receive details about the convicted person, their temporary release, the conditions of their release, and their expected release date, as well as information about life-sentence prisoners and the opportunity to give views when they are considered for final release.
$T5_{\text{ImpDistill}}$	In the case of life-sentence prisoners, the victim will be able to access the information about the minimum number of years the prisoner must serve, the date of the release and the opportunity to give their views.
Sentence	Using the wrong version of "too" doesn't automatically mean he's an ESL speaker, because too many native-speakers (from around the planet) get that wrong all the time, which I was politely (if sarcastically) pointing out.
PEGASUS (supervised)	the wrong version of "too" doesn't mean he's all speaker
T5 (supervised)	too many native speakers get that wrong time
GPT-3 (few-shot)	I was sarcastically pointing out that using the wrong version of "too" doesn't mean someone is an ESL speaker, as native speakers make this mistake often.
GPT-3 (zero-shot)	The speaker is pointing out that using the wrong version of "too" does not automatically mean someone is an ESL speaker, as native speakers from around the world make this mistake frequently.
$T5_{\text{ImpDistill}}$	Using the wrong version of "too" doesn't necessarily mean he's an ESL speaker, because a lot of native-speakers make that mistake.

Table 11: Example summaries from $T5_{\text{ImpDistill}}$, supervised baselines (PEGASUS and T5 fine-tuned on Gigaword), and unsupervised baselines (few-shot / zero-shot prompted GPT-3).

G Limitations

In this work, we limit our experiments to summarizing and paraphrasing a given sentence. In future works, IMPOSSIBLE DISTILLATION could be applied to a broader range of tasks, *e.g.* translation. To generate a parallel corpus for translation without human supervision, IMPOSSIBLE DISTILLATION could leverage the strong capability of recently-proposed multilingual LMs [39, 71] and cross-lingual filters [15]. Another direction would be to adapt IMPOSSIBLE DISTILLATION for longer input-output pairs, *e.g.* for paragraph-level summarization. A potential strategy here could be first generating the input article, then sequentially generating zero-shot summaries of the article with a fixed separator (*e.g.* `tl;dr`, [56]). As such, IMPOSSIBLE DISTILLATION could be extended to diverse range of tasks by re-defining the pair generation constraints and task-specific filters.

IMPOSSIBLE DISTILLATION makes use of a fixed set of filters (*e.g.* off-the-shelf NLI model) to determine which pair qualifies as a high-quality sample. Throughout the distillation pipeline, these filters remain frozen. Although our experiments show that the frozen filters are strong enough to distill a high-quality dataset than human-authored corpora, such filters may not always be accessible in wider range of tasks. Hence, future works could improve the framework by learning not only the task model that generates candidate pairs, but also the filter model that scores the plausibility of a given pair. We envision that by co-evolving the task model and filter model throughout the distillation stages, our framework could generalize to more complex problems such as commonsense reasoning, where it is non-trivial to define which pairs qualify as good task example.

H Human Evaluation Details

For human evaluation, we sample 300 sentences from XSUM, TL;DR and PubMed, then generate corresponding summaries from all methods. With an IRB approval, we recruit annotators from Amazon Mechanical Turk (MTurk), and ensure that all summaries are annotated by 3 distinct evaluators. To minimize subjectivity, we use 3-point Likert scale where annotators evaluate the fluency (whether the summary exhibits fleunt language), faithfulness (whether the summary well preserves the content of the original sentence and does not hallucinate), and conciseness (whether the summary is succinct enough) of each summary. We compensate workers with the hourly wage of $15.

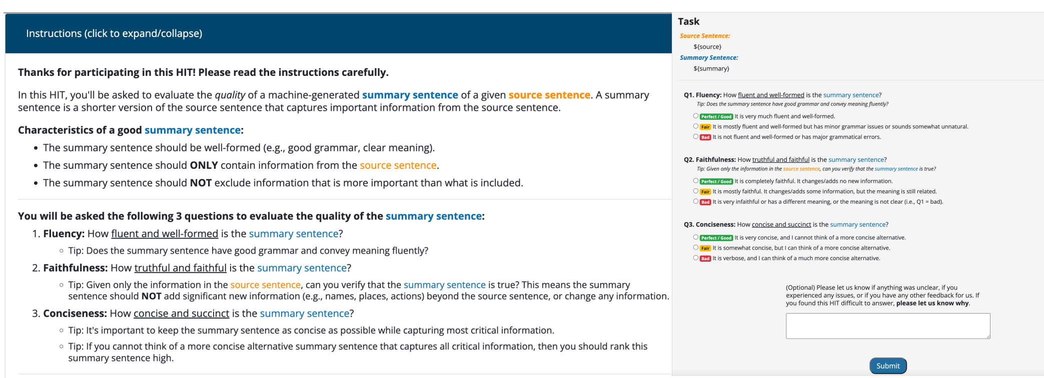

Figure 6: Screenshot of MTurk interface used for the human evaluation of model generated summaries.