Analyzing Multi-Task Learning for Abstractive Text Summarization

Frederic Kirstein[1,2,3], Jan Philip Wahle[1], Terry Ruas[1], Bela Gipp[1]
[1]Georg-August-Universität Göttingen, Germany
[2]Mercedes-Benz Group AG, Germany
[3]kirstein@gipplab.org

Abstract

Despite the recent success of multi-task learning and pre-finetuning for natural language understanding, few works have studied the effects of task families on abstractive text summarization. Task families are a form of task grouping during the pre-finetuning stage to learn common skills, such as reading comprehension. To close this gap, we analyze the influence of multi-task learning strategies using task families for the English abstractive text summarization task. We group tasks into one of three strategies, i.e., sequential, simultaneous, and continual multi-task learning, and evaluate trained models through two downstream tasks. We find that certain combinations of task families (e.g., advanced reading comprehension and natural language inference) positively impact downstream performance. Further, we find that choice and combinations of task families influence downstream performance more than the training scheme, supporting the use of task families for abstractive text summarization. Our code is publicly available [1].

1 Introduction

Self-supervised learning has been a significant success driver for generating high-quality abstractive summaries (Devlin et al., 2019; Liu et al., 2019b; Cohen and Gokaslan, 2020; Lewis et al., 2020; Raffel et al., 2020; Radford et al., 2019). Through self-supervision, language models implicitly learn intrinsic language features (e.g., syntax) from unlabeled data that they can use to solve downstream tasks (Brown et al., 2020). However, skills necessary to perform specific tasks often can be learned from an existing set of labeled data, requiring fewer training iterations (Rajpurkar et al., 2016; See et al., 2017). For example, to perform text summarization, a helpful skill is the ability to answer questions about texts (Rajpurkar et al., 2016).

The multi-task learning paradigm and its variations aim to acquire multiple skills simultaneously to succeed on the downstream tasks, e.g., T5 (Raffel et al., 2020), and are independent of a specific training stage (Aribandi et al., 2021). While studies on the effects of multi-task learning on a large scale exist (Aghajanyan et al., 2021; Sun et al., 2020; Aribandi et al., 2021) and are evaluated on broad natural language understanding benchmarks (Wang et al., 2019), they are lacking insight on the influence on abstractive text summarization. Furthermore, multi-task learning approaches are diverse in their methods (e.g., training scheme, mixing strategy, task families), hampering their comparison.

In this work, we investigate the role of multi-task learning on English abstractive text summarization. Therefore, we organize 18 pre-selected training tasks into six higher-level, modular task families. Further, we compare three training schemes for the pre-finetuning stage and their respective mixing strategies through changes of multiple scores.

Our experiments show that families' choice significantly impacts text summarization, while different training schemes have little influence. Moreover, pairing a text summarization task family with any other helps to stabilize the overall performance when transferring to unknown data. In some cases, we also found that a text summarization task family can be substituted by other family pairs, e.g., advanced reading comprehension and classification.

To summarize our contributions:

- We study the influence of multi-task learning by training models on six task families for the English abstractive text summarization task.
- We evaluate the co-training of different task families using statistical (e.g., ROUGE) and semantic metrics (e.g., BERTScore) for 18 datasets.

[1]https://github.com/FKIRSTE/GEM_emnlp2022-TOASTS

Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pages 54 - 77

- We compare the influence of three training schemes (i.e., sequential, simultaneous, continual multi-task learning) and two mixing strategies (i.e., proportional, equal).

2 Related Work

Multi-task learning and pre-finetuning. Transformers (Vaswani et al., 2017) such as BERT (Devlin et al., 2019) and GPT-3 (Brown et al., 2020) are trained using a two-step approach, the *pre-training* on large unlabeled corpora and the *finetuning* on a smaller, more specific (and usually labeled) downstream corpus. This bilateral approach allows language models to obtain general text representations once to perform many NLP downstream tasks with few gradient steps (e.g., document classification (Ostendorff et al., 2020a,b), plagiarism detection (Wahle et al., 2021, 2022b,c), media bias detection (Spinde et al., 2021, 2022)). However, pre-training is typically highly computationally expensive and requires dedicated ample infrastructure; few researchers can reproduce the pre-training of large language models. Therefore, recent works (Phang et al., 2018; Aghajanyan et al., 2021)) proposed additional training stages between pre-training and finetuning, i.e., *pre-finetuning*[2].

ERNIE 2.0 (Sun et al., 2020) proposes *continual multi-task learning*, in which tasks are trained incrementally, thereby building a queue of introduced tasks that re-appear throughout the training process, to counter catastrophic forgetting (McCloskey and Cohen, 1989; Kirkpatrick et al., 2017). MUPPET (Aghajanyan et al., 2021) and ExT5 (Aribandi et al., 2021) follow a *simultaneous* approach, drawing heterogeneous batches from multiple tasks and massively scale their training to >50 and >100 tasks respectively. MT-DNN (Liu et al., 2019a) organizes the prediction layer of a Transformer into four task families of common tasks of the GLUE benchmark (Wang et al., 2018) and learns each task *sequential* with their task order randomized. This study compares continual multi-task learning, simultaneous training, and sequential training for abstractive text summarization.

Task selection and relationship. Vu et al. (2020) conduct an empirical investigation on 33 tasks across three broad groups (i.e., text classification, question answering, and sequence labeling) to explore their inter- and intra-group training for different group sizes. Their experiments suggest that positive transfers between task groups are possible when the source dataset is small, and intergroup transfers are sensitive to group sizes. ExT5 (Aribandi et al., 2021) analyzes the correlation of task family representatives and shows, that summarization tasks (i.e., CNN/Daily Mail (See et al., 2017), XSum (Narayan et al., 2018), WikiLingua(Ladhak et al., 2020)) generally reduce performance on most other task families and that CBQA tasks (i.e., Natural Questions (Kwiatkowski et al., 2019), Trivia QA (Joshi et al., 2017), Hotpot QA (Yang et al., 2018)) are sensitive to multi-task learning. For the task relationship and transfer analysis, Aribandi et al. (2021) train on two families simultaneously and evaluate the first one. We expand the study of Aribandi et al. (2021) by adapting task families and respective representative tasks to be related to the text summarization task (Section 3.1), considering different family combinations, training approaches (Section 3.2), and tracking their performance through additional metrics for different unseen datasets (Section 4).

Multiple works leverage algorithms for the selection of training tasks, e.g., Ruder and Plank (2017) use Bayesian Optimization to learn similarity measures (i.e., Jensen-Shannon divergence (Lin, 1991) and Rényi divergence (Rényi et al., 1961)) and a Beta-Bernoulli multi-armed bandit with Thompson Sampling (Russo et al., 2018; Thompson, 1933) is used by AutoSem (Guo et al., 2019). Conversely, ExT5 (Aribandi et al., 2021) does not rely on automatic training task selection approaches as described by the preceding works and instead chooses an empirical approach to select tasks for higher-level task families. We follow the approach of Aribandi et al. (2021)'s task representative selection when choosing our tasks as the training task correlation analysis in ExT5 indicates which families could positively influence text summarization.

3 Methodology

We name our study TOASTS, a **T**ask-**O**riented **A**nalysi**S** for **T**ext **S**ummarization to investigate the effects of different task family combinations on English abstractive text summarization via a multi-task learning architecture. TOASTS groups selected pre-training tasks into task families and explores the correlation of these families, their influence on two downstream tasks, and their aggre-

[2]In this paper, we will use *intermediate training* and *pre-finetuning* interchangeably

Task Family	Task	Dataset	Source	Characteristics
Classification [CLS]	Sentiment Classification Sentiment Classification Topic Classification	GoEmotion (2020) IMDB (2011) AG News (2015)	Reddit IMDB ComeToMyHead	multi-label CLS binary CLS multi-class CLS
Commonsense [CMNS]	Fill-In-The-Blank Question Answering Question Answering	Winogrande (2021) PhysicaliQA (2019) SocialiQA (2019)	WSC dataset instructables.com crowdsourced	binary options binary options ternary options
Natural Language Inference [NLI]	Textual Entailment CLS Textual Entailment CLS Textual Entailment CLS	MNLI (2018) ANLI (2020) QNLI (2018)	SNLI corpus human-and-model-in-the-loop dataset Wikipedia	multi-label CLS multi-label CLS binary classification
Reading Comprehension [RC]	Binary QA Extractive QA Abstractive QA	BoolQ (2019) SQuAD (2016) TweetQA (2019)	Google Wikipedia Twitter	yes/no answer extractive answers abstractive answers allowed
Advanced RC [RC$^+$]	RC + Information Retrieval RC + Open Domain QA RC + CMNS	HotpotQA (2018) Natural Questions (2019) ReCoRD (2018)	Wikipedia Google, Wikipedia CNN/DailyMail and Internet Archive	multi-hop question answering answer information seeking questions extractive Machine RC
Summarization [SUM]	Extractive SUM Abstractive SUM Abstractive SUM	XSum (2018) WikiLingua [eng] (2020) AESLC (2019)	BBC WikiHow E-Mail	one-sentence summary one-sentence summary subject line generation

Table 1: Our selection of 18 representative datasets organized by their task family. For every dataset, we list the target task, the source, and the characteristics of the data. For a complete list of tasks, please see Appendix A.

gation through three training schemes. Therefore, we use pre-finetuning, a second inexpensive pre-training stage between pre-training and fine-tuning, which was recently proposed by Muppet (Aghajanyan et al., 2021) and tested by ExT5 (Aribandi et al., 2021). Pre-finetuning has two main parts: the *task family setup* and the *training strategies*. The task family setup groups different tasks and related datasets into broader families according to their primary objective. The tasks of these families are then combined following a training strategy and evaluated into a final task. Figure 1 illustrates the components of TOASTS, which are detailed in the following sections.

3.1 Task family setup

Selection. A myriad of NLP downstream tasks (e.g., word sense disambiguation and paraphrase detection) can be considered when choosing a multi-task architecture. Without computational limits, one could explore all possible permutations of tasks and the influence of the respective tasks on downstream performance. Unfortunately, as the number of tasks grows by more than their factorial number, joint training becomes computationally prohibitive (Aribandi et al., 2021). Therefore, we organize tasks into six high-level families (Aribandi et al., 2021; Brown et al., 2020) and perform combinations on their family levels: *classification (CLS)*, *commonsense reasoning (CMNS)*, and *natural language inference (NLI)*, *reading comprehension (RC)*, *advanced reading comprehension* [3] *(RC$^+$)*, *summarization (SUM)*. We compose each task family of three datasets that tackle different aspects of the problem, as shown in Table 1.

The selected tasks in TOASTS should not be seen as an exhaustive list of all NLP downstream tasks; instead, they should be considered an educated selection to measure task family influence on text summarization. An extended list of planned tasks for future analyses can be found in Table 7 in Appendix A.

Task mixing. After pre-selecting representative tasks for each family, we control the percentage of data ingested from each task using a task mixing strategy. We consider two methods for processing all combinations of task families: *proportional mixing* (Sanh et al., 2019; Aribandi et al., 2021) and *equal mixing* (Raffel et al., 2020). Equal mixing picks training samples from each task with equal probability, while proportional mixing sets the probability to the proportion of each task's size. The use of proportional mixing as a default strategy is the recommended approach for various multi-task learning strategies (Sanh et al., 2019). However, continual multi-task learning (Section 3.2) requires an equal mixing strategy even though related studies have shown it to be sub-optimal (Raffel et al., 2020). While we sample either proportional or equal within task families, we draw equal between task families to balance the influence of potentially different task families. We leave to future

[3] Aribandi et al. (2021) refer to this family as Closed Book Question Answering (CBQA).

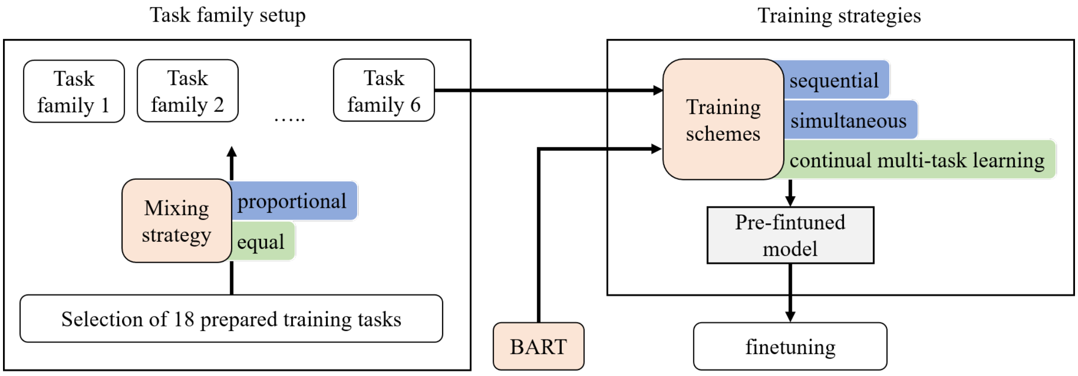


Figure 1: The central architecture of TOASTS. The intermediate training phase commences the **task family setup** (left) by organizing the pre-selected training tasks into families of similar problems and applying two (proportional, equal) intra-family mixing strategies. The **training strategies** (right) continue by processing and organizing the generated task families into batches according to one of three training schemes (sequential, simultaneous, continual multi-task learning). After pre-finetuning BART, the resulting model is finetuned and evaluated on two abstractive text summarization datasets (Reddit TIFU, arXiv). The training/mixing scheme pairings are marked by the background colors green and blue.

work the investigation of the effects of different amounts of tasks and samples per family.

3.2 Training strategies

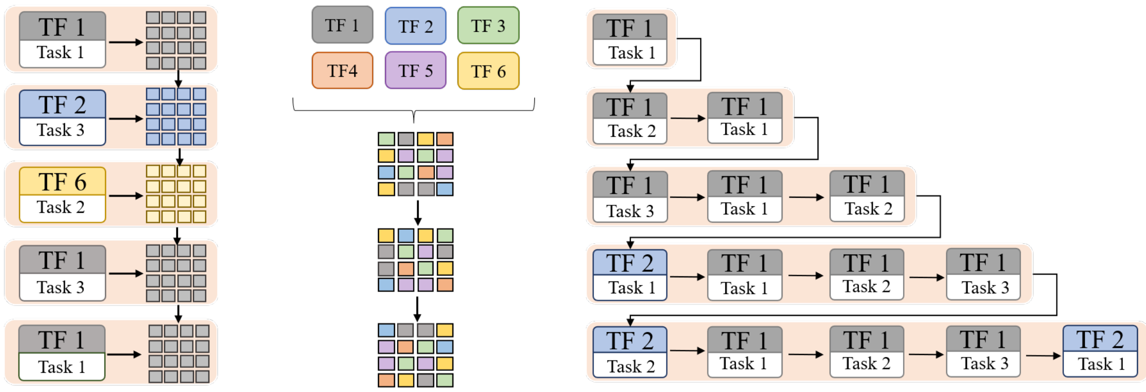
Training Schemes. Multi-task learning during a pre-finetuning stage allows us to start from a pre-trained checkpoint, decreasing the final task's overall cost. We explore three multi-task learning training schemes for the pre-finetuning as Figure 2 shows: *sequential learning* (seq) (McCloskey and Cohen, 1989; Biesialska et al., 2020), *simultaneous learning* (sim) (Caruana, 1997; Aghajanyan et al., 2021), and *continual multi-task learning* (cMTL) (Sun et al., 2020). In the *sequential* approach, training batches are composed of a single dataset, i.e., homogeneous batches, and their processing order is sequentially randomized (Liu et al., 2019a). This approach achieves a concentrated task learning on the batch level while keeping the overall variety, therefore learning a task more thoroughly before moving to the next. For the *simultaneous* strategy, we combine all tasks into a single pool and draw randomly from it (Aghajanyan et al., 2021; Aribandi et al., 2021). This prominent approach introduces task variety on the batch level by constantly challenging the model with different approaches, forcing it to identify intrinsic commonality between the task families quickly. For *continual multi-task learning*, we adjust the concept of ERNIE 2.0 (Sun et al., 2020) to adapt it to our task family configuration. As our tasks corpus is less extensive than the training dataset used in ERNIE 2.0, we have to rejig the number of stages and training steps in TOASTS. Therefore, when including new tasks and task families, we change their total number of steps to 9k, and 27k, respectively, as Table 2 shows. One difference from ERNIE 2.0 is that once a new task is introduced to the pipeline and trained for the first time at timestep t, we move it to the end of the queue of previously trained tasks as the last one to be executed in $t + 1$. Using the order in (Sun et al., 2020) as an alternative way of including and carrying new tasks, yields worse results (Table 8). Through the pre-determined task order of this approach, we can control which task families follow each other and how fundamental a task is by introducing it earlier than others.

Task	**S1**	**S2**	**S3**	**S4**	**S5**	...	**S18**
TF1.1	500	500	500	500	500	...	500
TF1.2	-	1k	500	500	500	...	500
TF1.3	-	-	1.5k	500	500	...	500
TF2.1	-	-	-	2k	500	...	500
TF2.2	-	-	-	-	2.5k	...	500
...	-	-	-	-	-	...	500
TF6.3	-	-	-	-	-	-	9k

Table 2: The number of batches during cMTL training depends on the training stage and the number of introduced tasks. S1 to S16 denote the stages when a new task TF1.1 to TF6.3 is introduced. TF1.1 indicates the first task of task family 1, TF1.2 the second task of task family 1 etc.

4 Experimental setup

Model. For all experiments, we use BART-Large (Lewis et al., 2020) to probe combinations of task families, mixing, and training strategies in



(a) Sequential learning. (b) Simultaneous learning. (c) Continual multi-task learning.

Figure 2: TOASTS's three training strategies. (a) Sequential learning (seq) draws a batch with samples from one task of a task family at a time for every training stage. The order of tasks is randomized. (b) Simultaneous learning (sim) samples from all available tasks at the same time. (c) Continual multi-task learning (cMTL) introduces a new task in each training stage, which is added to the end of the training queue.

TOASTS. BART is a two-stage denoising autoencoder that corrupts its input text and reconstructs it through a sequence-to-sequence model. We chose BART because of its ability to perform a wide range of downstream tasks, such as paraphrase detection (Wahle et al., 2022b), fake news identification (Wahle et al., 2022a), and text summarization (Lewis et al., 2020). Additionally, in our preliminary experiments, BART also performed better than other candidate models such as PEGASUS (Zhang et al., 2020) and T5 (Raffel et al., 2020) (comparison in Tables 9 and 10 in appendix B).

Tokenization. We tokenize text using the BART-Large tokenizer and augment all texts to include task-specific prompts such as `'question:'` or `'context:'`. Further, we structure the samples to follow a uniform text-to-text style which allows the model to handle multi-task learning across different task families without needing task-specific losses, loss scaling or explicit gradient accumulation on heterogeneous batches (Liu et al., 2019a; Aghajanyan et al., 2021).

Hyperparameters. We run our experiments on 8 NVIDIA A100s with a total of 320GB GPU memory. The models are trained with a total batch size of 8 for three epochs and up to 60k global steps for six task families during pre-finetuning (finetuning: 16k for Reddit TIFU, 70k for arXiv) with half-precision (fp16). The pre-finetuning takes between 17min (single task family) and 11h (all task families). The finetuning takes 2.2h for Reddit TIFU and 19.85h for arXiv. During pre-finetuning, we set the input sequence to 512 tokens and the target sequence to 128 as a compromise for training time and context. During finetuning, the sequence lengths are increased to 1024 and 512 for input and target, respectively, to capture the full context of both evaluation datasets. For other hyperparameters we refer the reader to Table 41 in Appendix D.

Evaluation. To understand each task family's influence, mixing, and training strategies, we evaluate the text summarization task using two datasets: Reddit TIFU (Kim et al., 2019) and arXiv (Cohan et al., 2018). Reddit TIFU is composed of 120K posts from online conversations, with the task of creating a tldr[4] summary from the post. The arXiv dataset consists of 250K scientific articles with the task of deriving the abstract from the full text. These datasets are commonly referred to as challenging abstractive summarization tasks (Zhang et al., 2020; He et al., 2020). In combination, they provide a balanced landscape as Reddit TIFU contains shorter examples with an average of 432 words per post and 23 per summary, relying on simpler linguistic, and arXiv longer examples with 4938 words per document and 220 per summary constructed from elaborated text.

During our experiments, we consider a combination of *count-based* and *semantic* metrics to assess the quality of produced summaries. We use BLEU (Papineni et al., 2002), ROUGE (1, 2, L) (Lin, 2004), and METEOR (Banerjee and Lavie, 2005), which favor precision, recall, and harmonic mean, respectively. Even though these traditional metrics can work well for similarly worded sum-

[4] too long; didn't read

Task Families	Reddit TIFU			arXiv		
	seq	sim	cMTL	seq	sim	cMTL
CLS	0.226	0.233	0.060	0.154	0.287	**0.286**
CMNS	0.226	0.078	0.078	0.286	0.197	0.163
NLI	0.030	0.082	0.082	0.168	0.111	0.182
RC	0.230	**<u>0.235</u>**	0.230	0.282	0.284	0.282
RC^{+}	0.224	0.082	0.078	0.282	**<u>0.289</u>**	0.203
SUM	**0.231**	**<u>0.235</u>**	**0.231**	**0.288**	0.282	**0.286**
ALL	0.222	0.228	0.037	0.281	0.279	0.008
BART (baseline)	$0.087^{\dagger}$	$0.087^{\dagger}$	$0.087^{\dagger}$	$0.281^{\dagger}$	$0.281^{\dagger}$	$0.281^{\dagger}$

Table 3: Results (METEOR) for single task families and the combination of all task families for the Reddit TIFU and arXiv datasets. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent from training. †Repeated result for baseline without training scheme.

maries, they are limited when wording changes, but the semantic meaning remains the same (Bhandari et al., 2020; Huang et al., 2021). To assess semantic similarity better, we also include BERTScore (Zhang et al., 2019a), a similarity measure that maximizes the cosine similarity between candidate and reference contextualized token embeddings via BERT (Devlin et al., 2019) in a greedy manner.

4.1 Experimental results and discussion

We structure our experiments into four research questions, which tackle the relevance of task families and dataset compatibility (RQ1), the effects of co-training text summarization task families with other families (RQ2), the co-training of task families excluding text summarization (RQ3), and the co-training of text summarization and two different task families (RQ4).

We pre-finetune our baseline model (BART-Large) for each experiment on specific task families (e.g., CLS, CMNS) and evaluate the resulting models into the Reddit TIFU and arXiv datasets. Tables 3 to 6 show the different task mixing and training strategies. Sequential (seq) and simultaneous (sim) training strategies use proportional mixing, while continual multi-task learning (cMTL) uses equal mixing. Because of space constraints, we report our results only for the METEOR metric, which proved to be the most sensitive to our experiments. We include a complete list of results for BertScore, BLEU, METEOR, and ROUGE (1, 2, L) in Appendices C.1 and C.2.

RQ1*: Does increasing the number of pre-finetuning datasets increase downstream task performance for text summarization?*

A. To identify if the text summarization downstream task benefits from unconstrained usage of multiple task families, we compare how each task family performs against the combination of all.

As Table 3 shows, the SUM task family consistently outperforms the combination of all families for both datasets (followed by RC), except for the sim training scheme on arXiv. The increase in performance through pre-training SUM is somehow expected, as it is the most related task family to the actual problem, i.e., abstractive text summarization. Conversely, NLI performs the worst when compared to any other task family. Pre-finetuning generally positively affects BART compared to its baseline, except for a few cases (e.g., cMTL-RC^{+}, NLI). Overall, the sim training strategy greatly influenced downstream task performance.

Our results suggest that combining all task families is suboptimal for text summarization, which challenges recent observations for other NLP tasks (Aghajanyan et al., 2021; Aribandi et al., 2021). Also, increasing the number of task families requires high compute budgets. As we train each task family individually or all simultaneously, it is unclear how much influence a summarization task family (e.g., SUM) has on the others.

RQ2*: How much does the text summarization task affect other task families?*

A. As SUM is closely related to the text summarization task, and it yields the best results in RQ1, we explore how its combination with another task family affects the resulting model. Table 4 shows the results of combining SUM with other task families. Aside from a few cases (e.g., arXiv sim for SUM+RC^{+}), pairing with the SUM family improves over almost every single run in Table 3 and the combination of all task families.

Task Families	Reddit TIFU			arXiv		
	seq	sim	cMTL	seq	sim	cMTL
SUM+CLS	0.230	**0.233**	0.077	**0.285**	0.285	0.283
SUM+CMNS	0.232	0.231	<u>**0.234**</u>	0.153	0.286	<u>**0.288**</u>
SUM+NLI	0.223	**0.233**	0.223	0.282	**0.287**	0.282
SUM+RC	**0.233**	0.229	<u>**0.234**</u>	**0.285**	0.280	0.283
SUM+RC+	0.230	0.225	<u>**0.234**</u>	0.284	0.281	0.284
BART (baseline)	0.087†	0.087†	0.087†	0.281†	0.281†	0.281†

Table 4: Results (METEOR) for the combination of SUM and different task families for the Reddit TIFU and arXiv datasets. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training. †Repeated result for baseline without training scheme.

While some task families' combinations obtain small benefits (seq-SUM+RC), others are greatly affected (e.g., cMTL-SUM+CMNS) for both datasets. The BART baseline performs better than the pre-finetuning in only two cases, i.e., SUM+CLS for Reddit TIFU (cMTL) and SUM+CMNS for arXiv (seq). We observe fewer outliers with low scores when pairing SUM with other task families than in RQ1. Individual training improved the performance on arXiv the most (seq and sim), while for Reddit TIFU, the combination of task families was more effective (seq and cMTL).

Low scores are also less frequent when combining task families with one exception, i.e., cMTL-SUM+CLS for Reddit TIFU. The lowest scores in RQ1 (e.g., NLI, CMNS) and RQ2 (CLS) might be related to the fact that these tasks are not contributing to the learned weights of the downstream task. As Reddit TIFU uses mostly informal language and its input sequence and summaries are short, this might justify these low scores.

The improvements in Table 4 over the BART baseline are likely to be related to the SUM family rather than a mixing strategy or training scheme. The results of individually training the SUM family (RQ1) are equal or marginally higher when combined with other task families (e.g., 0.233 for SUM+RC vs. 0.231 SUM). As the SUM family seems to substantially impact co-training multiple tasks, we are interested in evaluating the influence of families other than SUM.

***RQ3**: How do non text summarization task families influence each other?*

A. We remove the SUM family and co-train all possible pairs of task families. Table 5 shows that the co-training of non text summarization task families (e.g., NLI+RC+) can achieve equal or better results in comparison to single SUM training (Table 3) or its combination with other task families (Table 4) for both Reddit TIFU and arXiv. Other combinations such as CLS+RC and RC+RC+ also achieve strong results.

Conversely, the combination of task families with good results individually seems to have a harmful influence on each other when paired. While CLS and CMNS have good results individually (0.226 and 0.226 for the seq strategy on Reddit TIFU), their pairing (e.g., CLS+CMNS) is strongly negative (e.g., 0.078 for the seq strategy on Reddit TIFU). As in Table 3, different training schemes seem to be a less dominant factor than task family choice during pre-finetuning. Therefore, a proper task family combination should precede architectural training options.

Our results suggest that non text summarization task families can be used to substitute for the SUM family. Specifically, all best-performing results include RC or RC+ in their configuration. A possible explanation for the stark influence of RC/RC+ is that their problem of understanding texts is closely related to summarizing texts. A link between reading comprehension and text summarization is also observed by psychologists in various studies (e.g., Cohen (2006); Kintsch and van Dijk (1978); Yu (2008)).

***RQ4**: How are non text summarization task family pairs affected by SUM?*

A. Considering the positive effect of SUM in other families (RQ2), we investigate its influence in task family pairs (RQ3) as Table 6 shows. For this research question, we only consider Reddit TIFU as it provides a more challenging scenario (i.e., informal, short texts) and limits our computational

Task Families	Reddit TIFU			arXiv		
	seq	sim	cMTL	seq	sim	cMTL
CLS+CMNS	0.078	0.078	0.060	0.078	0.050	0.162
CLS+NLI	0.077	0.077	0.046	0.050	0.003	0.276
CLS+RC	0.231	0.231	0.230	**0.287**	0.283	0.181
CLS+RC+	0.229	0.229	0.082	0.284	0.288	**0.287**
CMNS+NLI	0.231	0.231	0.081	0.137	0.212	0.118
CMNS+RC	0.227	0.227	0.077	0.283	0.284	0.179
CMNS+RC+	0.232	0.232	**0.232**	0.279	0.280	0.082
NLI+RC	0.231	0.231	0.231	0.285	0.285	0.284
NLI+RC+	**0.233**	<u>**0.234**</u>	0.227	0.286	<u>**0.290**</u>	0.282
RC+RC+	0.228	0.228	0.228	**0.287**	0.281	0.285
BART (baseline)	0.087†	0.087†	0.087†	0.281†	0.281†	0.281†

Table 5: Results (METEOR) for the combination of all pairs of task families (except for SUM) for the Reddit TIFU and arXiv datasets. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training. †Repeated result for baseline without training scheme.

Task Families	Reddit TIFU		
	seq	sim	cMTL
SUM+CLS+CMNS	0.228	0.227	0.077
SUM+CLS+NLI	0.231	0.231	0.082
SUM+CLS+RC	**0.235**	0.228	0.229
SUM+CLS+RC+	**0.235**	0.233	0.082
SUM+CMNS+NLI	0.230	<u>**0.236**</u>	0.229
SUM+CMNS+RC	0.234	0.232	**0.230**
SUM+CMNS+RC+	0.232	0.231	0.228
SUM+NLI+RC	0.229	0.231	0.228
SUM+NLI+RC+	0.234	0.229	0.229
SUM+RC+RC+	0.227	0.234	0.228
BART (baseline)	0.087†	0.087†	0.087†

Table 6: Results (METEOR) for the combination of all pairs of task families and SUM for Reddit TIFU. Values in **bold** represent the highest results for a training scheme. <u>Underline</u> values are the highest results for that dataset independent of training. †Repeated result for baseline without training scheme.

budget (family co-training is increasingly expensive when the number of task families grows).

Including SUM mitigates the adverse effects of combining CLS+CMNS (e.g., 0.228 vs. 0.078 for the seq training scheme) and CLS+NLI (e.g., 0.231 vs. 0.077 for the seq training scheme), except for the cMTL training scheme. However, the scores for CLS+RC+ are almost unchanged. The seq and sim training schemes still perform best (e.g., CMNS+NLI) but for different task family combinations compared to the previous research questions' results (e.g., NLI+RC+ in RQ3). For the best performing task families pairs in RQ3, only CLS+RC and CLS+RC+ are still the top results when including SUM. As in Table 4, the SUM family seems to provide stability to the results, as we see fewer fluctuations than in Table 5. We assume the stability provided by SUM would also be present in the inclusion of more task families. Further, we observe the positive influence of RC and RC+ when pairing three task families excluding SUM (Tables 26 to 28).

5 Conclusion & Future Work

In this work, we studied the influence of multi-task learning combinations of task families during the pre-finetuning stage for English abstractive text summarization. We trained three different training strategies, six task families composed of 18 tasks, and evaluated two downstream tasks.

Our experiments show that non text summarization task families, e.g., advanced reading comprehension, can be used as a substitute for the summarization task (RQ2) or the combination of all task families (RQ1). However, including the summarization task family in the training process positively impacts the downstream performance compared to non text summarization family combinations. Further, our analysis shows that training strategies have little influence on the overall performance compared to the task family selection.

We see this analysis as the first step to understanding training strategies and task families for text summarization. In the future, we want to investigate more tasks (both in number and diversity) per task family, training schemes, and mixing strategies. We also plan to include psychological studies comparing the similarities of textual understanding tasks as a starting point for task family preselection.

Limitations

With the organization of tasks and datasets into task families, this study highly depends on these representative tasks' domain and expressiveness. As Aribandi et al. (2021) faced similar problems, we followed their guidance to select representatives to consist of a diverse set of datasets to train and evaluate on and to partition task families as mutually exclusive as possible while being related to abstractive text summarization. However, none of the datasets are perfectly isolated and can only be used as a proxy for a larger task family.

Ethical Considerations

This study depends on existing resources and generative models; thus, it is not free of biases and possible ethical considerations. One problem is the generation of text summaries that contain non-factual information, meaning distortion, social biases such as political stances, or abusive language (Gooding, 2022). To mitigate these problems we plan to condition the generation of trained models for unsafe content or other harmful text to return an empty string.

Furthermore, TOASTS is licensed to the public under a copyright policy that allows unlimited reproduction, distribution, and hosting on any website or medium. Hence, anyone can exploit its limitations and inherited biases to propagate and amplify unintentional societal problems.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with prefinetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Prakash Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. 2021. Ext5: Towards extreme multi-task scaling for transfer learning. *CoRR*, abs/2111.10952.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. Ms marco: A human generated machine reading comprehension dataset.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, 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.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. *CoRR*, abs/1903.04561.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Rich Caruana. 1997. Multitask learning. *Mach. Learn.*, 28(1):41–75.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short*

Papers), pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Andrew D. Cohen. 2006. The coming of age of research on test-taking strategies. *Language Assessment Quarterly*, 3(4):307–331.

Vanya Cohen and Aaron Gokaslan. 2020. Opengpt-2: Open language models and implications of generated text. *XRDS*, 27(1):26–30.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. Eraser: A benchmark to evaluate rationalized nlp models.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proc. of NAACL*.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. *Computer Speech & Language*, 59:123–156.

Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Association for Computational Linguistics.

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

Sian Gooding. 2022. On the ethical considerations of text simplification. In *Ninth Workshop on Speech and Language Processing for Assistive Technologies (SLPAT-2022)*, pages 50–57, Dublin, Ireland. Association for Computational Linguistics.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2019. AutoSeM: Automatic task selection and mixing in multi-task learning. 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 3520–3531, Minneapolis, Minnesota. Association for Computational Linguistics.

Junxian He, Wojciech Kryściński, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2020. Ctrlsum: Towards generic controllable text summarization.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning.

Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021. The factual inconsistency problem in abstractive text summarization: A survey. *arXiv preprint arXiv:2104.14839*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface:a challenge set for reading comprehension over multiple sentences. In *NAACL*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of Reddit posts with multi-level memory networks. 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 2519–2531, Minneapolis, Minnesota. Association for Computational Linguistics.

Walter Kintsch and Teun A. van Dijk. 1978. Toward a model of text comprehension and production. *Psychological Review*, 85(5):363–394.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Vid Kocijan, Thomas Lukasiewicz, Ernest Davis, Gary Marcus, and Leora Morgenstern. 2020. A review of winograd schema challenge datasets and approaches.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online. Association for Computational Linguistics.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2019. Commongen: A constrained text generation challenge for generative commonsense reasoning.

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

J. Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

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

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

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. 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*, Brussels, Belgium.

Malte Ostendorff, Terry Ruas, Till Blume, Bela Gipp, and Georg Rehm. 2020a. Aspect-based Document Similarity for Research Papers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6194–6206, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Malte Ostendorff, Terry Ruas, Moritz Schubotz, Georg Rehm, and Bela Gipp. 2020b. Pairwise Multi-Class Document Classification for Semantic Relations between Wikipedia Articles. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, pages 127–136, Virtual Event China. ACM.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

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

Jason Phang, Thibault Févry, and Samuel R. Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *CoRR*, abs/1811.01088.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2018. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations.

Adam Poliak. 2020. A survey on recognizing textual entailment as an nlp evaluation.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

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(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Alfréd Rényi et al. 1961. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1. Berkeley, California, USA.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with Bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization.

Daniel J. Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. 2018. A tutorial on thompson sampling. *Found. Trends Mach. Learn.*, 11(1):1–96.

Amrita Saha, Vardaan Pahuja, Mitesh M. Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Timo Spinde, Jan-David Krieger, Terry Ruas, Jelena Mitrović, Franz Götz-Hahn, Akiko Aizawa, and Bela Gipp. 2022. Exploiting Transformer-Based Multitask Learning for the Detection of Media Bias in News Articles. In Malte Smits, editor, *Information for a Better World: Shaping the Global Future*, volume 13192, pages 225–235. Springer International Publishing, Cham.

Timo Spinde, Manuel Plank, Jan-David Krieger, Terry Ruas, Bela Gipp, and Akiko Aizawa. 2021. Neural Media Bias Detection Using Distant Supervision With BABE - Bias Annotations By Experts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1166–1177, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8968–8975.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge.

William R Thompson. 1933. On the Likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. 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 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926, Online. Association for Computational Linguistics.

Jan Philip Wahle, Nischal Ashok, Terry Ruas, Norman Meuschke, Tirthankar Ghosal, and Bela Gipp. 2022a. Testing the Generalization of Neural Language Models for COVID-19 Misinformation Detection. In Malte Smits, editor, *Information for a Better World: Shaping the Global Future*, volume 13192, pages 381–392. Springer International Publishing, Cham.

Jan Philip Wahle, Terry Ruas, Tomáš Foltýnek, Norman

Meuschke, and Bela Gipp. 2022b. Identifying Machine-Paraphrased Plagiarism. In Malte Smits, editor, *Information for a Better World: Shaping the Global Future*, volume 13192, pages 393–413. Springer International Publishing, Cham.

Jan Philip Wahle, Terry Ruas, Frederic Kirstein, and Bela Gipp. 2022c. How large language models are transforming machine-paraphrased plagiarism. *arXiv preprint arXiv:2210.03568*.

Jan Philip Wahle, Terry Ruas, Norman Meuschke, and Bela Gipp. 2021. Incorporating Word Sense Disambiguation in Neural Language Models. *arXiv:2106.07967 [cs]*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

Adina Williams, Tristan Thrush, and Douwe Kiela. 2020. Anlizing the adversarial natural language inference dataset.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. TWEETQA: A social media focused question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5020–5031, Florence, Italy. Association for Computational Linguistics.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. 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)*. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Guoxing Yu. 2008. Reading to summarize in english and chinese: A tale of two languages? *Language Testing*, 25(4):521–551.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *arXiv:1912.08777 [cs]*.

Rui Zhang and Joel Tetreault. 2019. This email could save your life: Introducing the task of email subject line generation.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019a. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2019b. Semantics-aware bert for language understanding.

A Tasks and Families

Table 7 shows an extended version of pre-finetuning tasks in Table 1 to-be-considered in future work

B Additional Models

Tables 8 to 10 shows the results for different models and loop orders. BART performed best compared to models from related work, which is why we chose the model throughout our experiments.

C Extended Results

C.1 Extended Results on Reddit TIFU

Tables 13 to 27 show the detailed evaluation for each research question and all tested combinations of task families evaluated on the Reddit TIFU datasets. The tables are divided according to their training scheme, i.e., each table shows one of the three training schemes (sim, seq, cMTL).

C.2 Extended Results on arXiv

Tables 31 to 39 show the detailed evaluation for each research question and all tested combinations of task families evaluated on the arXiv datasets. The tables are divided according to their training scheme, i.e., each table shows one of the three training schemes (sim, seq, cMTL).

D Hyperparameters

Table 41 shows the hyperparameters used throughout the pre-finetuning and finetuning experiments.

TF	Task	Dataset	Citation
CLS	Topic Classification	AG News	(Zhang et al., 2015)
	Text Classification	Civil Comments	(Borkan et al., 2019)
	Text Classification	FEVER	(Thorne et al., 2018)
	Emotion Classification	GoEmotions	(Demszky et al., 2020)
	Sentiment Classification	IMDB	(Maas et al., 2011)
	Sentiment Classification	Rotten Tomatoes	(Pang and Lee, 2005)
	Text Classification	Trec	(Li and Roth, 2002; Hovy et al., 2001)
	classification	Word-in-Context	(Pilehvar and Camacho-Collados, 2018)
	Sentiment Classification	Yelp Polarity	(Zhang et al., 2015)
	Linguistic Acceptability	CoLA	(Warstadt et al., 2018)
	Sentiment Classification	SST-2	(Socher et al., 2013)
CMNS	Open Domain QA	AI2 Reasoning (Challenge ARC)	(Yadav et al., 2019)
	Concepts to Text Generation	CommonGen (CG)	(Lin et al., 2019)
	Sequential Question Answering	CQA	(Saha et al., 2018)
	Commonsense Inference	HellaSWAG	(Zellers et al., 2019)
	Question Answering	PhysicaliQA	(Bisk et al., 2019)
	Question Answering	SocialiQA	(Sap et al., 2019)
	Text Classification	SWAG	(Zellers et al., 2018)
	Fill-In-A-Blank	WinoGrande	(Sakaguchi et al., 2021)
	Question Answering	Winograd Scheme (Challenge)	(Kocijan et al., 2020)
	Open-Domain-QA	CommonSense QA	(Talmor et al., 2018)
NLI	Textual Entailment Classification	ANLI (Adverserial NLI)	(Williams et al., 2020)
	Natural Language Inference	HANS	(McCoy et al., 2019)
	Textual Entailment Classification	MNLI	(Williams et al., 2018)
	Textual Entailment Classification	QNLI	(Wang et al., 2018)
	Textual Entailment Classification	RTE	(Poliak, 2020)
	Textual Entailment Classification	SciTail	(Khot et al., 2018)
	Natural Language Inference	SNLI	(Zhang et al., 2019b)
	Natural Language Inference	WNLI	(Wang et al., 2018)
RC	Binary QA	BoolQ	(Clark et al., 2019)
	Multiple Choice QA	Cosmos QA	(Huang et al., 2019)
	Multi-Sentence QA	Eraser Multi RC	(DeYoung et al.; Khashabi et al., 2018)
	Extractive QA	SQUAD	(Rajpurkar et al., 2016)
	Extractive QA	TriviaQA	(Joshi et al., 2017)
	Abstractive QA	TweetQA	(Xiong et al., 2019)
	Multiple Choice QA	RACE	(Lai et al., 2017)
RC^+	Text2Text Generation	E2E	(Dušek et al., 2020)
	RC + Question Answering	MSMarco	(Bajaj et al., 2016)
	RC + Open Domain QA	Natural Questions	(Kwiatkowski et al., 2019)
	RC + Commonsense Reasoning	RECORD	(Zhang et al., 2018)
	RC + Information Retrieval	HotpotQA	(Yang et al., 2018)
	RC + Extractive QA	DROP	(Dua et al., 2019)
SUM	Abstractive Summarization	Aeslc	(Zhang and Tetreault, 2019)
	Extractive Summarization	Billsum	(Eidelman, 2019)
	Abstractive Summarization	CNN	(See et al., 2017; Hermann et al., 2015)
	Headline Generation	Gigaword	(Rush et al., 2015)
	Abstractive Summarization	Multinews	(Fabbri et al., 2019)
	Abstractive Summarization	WikiLingua [eng]	(Ladhak et al., 2020)
	Extractive Summarization	XSUM	(Narayan et al., 2018)

Table 7: An extended list of Table 1. This list can be used to extend TOASTS to more tasks and datasets in future work. TF stands for Task Family.

order	BERTScore	BLEU	METEOR	ROUGE-1	ROURGE-2	ROUGE-L
ascending (ours)	0.881	0.057	0.229	0.284	0.096	0.228
descending	0.861	0.003	0.082	0.095	0.012	0.085

Table 8: Results of different loop orders tested. Let t denote the current training stage, then the ascending order for the training stage t is Task_t, Task_1, Task_2, ..., Task_{t-1}. The descending order follows for the same training stage t the form Task_t, Task_{t-1}, Task_{t-2}, ..., Task_1.

model	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L	Time
BART	**0.881**	**0.061**	**0.231**	**0.286**	**0.100**	**0.233**	0.75h
T5	**0.881**	0.052	0.218	0.282	0.090	0.229	1.15h
PEGASUS	0.876	0.058	0.215	0.264	0.094	0.216	1h

Table 9: Results of different models used. The models were finetuned on Reddit TIFU without pre-finetuning and with full precision. Values in **bold** represent the highest results for a training scheme.

model	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L	Time
BART	**0.864**	**0.129**	**0.306**	**0.444**	**0.168**	0.267	13.5h
T5	**0.864**	0.120	0.291	0.416	0.153	**0.272**	27.5h
PEGASUS	0.858	0.122	0.291	0.414	0.148	0.253	18.5h

Table 10: Results of different models used. The models were finetuned on arXiv without pre-finetuning and with full precision. Values in **bold** represent the highest results for a training scheme.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS	0.881	0.057	0.226	0.282	0.097	0.229
CMNS	0.881	0.055	0.226	0.282	0.095	0.228
NLI	0.869	0.000	0.030	0.088	0.006	0.083
RC	<u>**0.882**</u>	0.057	0.230	0.285	**0.098**	0.230
RC+	0.881	0.056	0.224	0.281	0.096	0.229
SUM	0.881	**0.061**	**0.231**	**0.287**	**0.098**	**0.231**
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 11: RQ1 results (single task family) for Reddit TIFU and the sequential strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS	0.881	0.061	0.233	0.286	0.099	0.232
CMNS	0.863	0.003	0.078	0.091	0.013	0.081
NLI	0.863	0.003	0.082	0.095	0.012	0.085
RC	0.881	0.061	**0.235**	<u>**0.290**</u>	0.100	0.232
RC+	0.863	0.003	0.082	0.095	0.012	0.085
SUM	<u>**0.882**</u>	**0.062**	**0.235**	0.288	**0.102**	**0.234**
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 12: RQ1 results (single task family) for Reddit TIFU and the simultaneous strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS	0.853	0.002	0.060	0.095	0.012	0.085
CMNS	0.863	0.003	0.078	0.091	0.013	0.081
NLI	0.863	0.003	0.082	0.095	0.012	0.085
RC	**0.881**	**0.059**	0.230	**0.287**	**0.098**	0.231
RC^+	0.863	0.003	0.078	0.091	0.013	0.080
SUM	**0.881**	**0.059**	**0.231**	**0.287**	**0.098**	**0.232**
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 13: RQ1 results (single task family) for Reddit TIFU and the continual multi-task learning strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
ALL	**0.880**	**0.053**	**0.222**	**0.278**	**0.092**	**0.225**
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 14: RQ1 results (all task families) for Reddit TIFU and the sequential strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
ALL	**0.881**	**0.057**	**0.228**	**0.283**	**0.095**	**0.228**
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 15: RQ1 results (all task families) for Reddit TIFU and the simultaneous strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
ALL	0.819	0.000	0.037	0.000	0.000	0.000
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 16: RQ1 results (all task families) for Reddit TIFU and the continual multi-task learning strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
SUM + CLS	0.881	**0.061**	0.230	0.284	0.098	0.230
SUM + CMNS	0.881	0.060	0.232	0.287	0.098	0.231
SUM + NLI	0.881	0.053	0.223	0.280	0.094	0.225
SUM + RC	**<u>0.882</u>**	**0.061**	**0.233**	**0.288**	**<u>0.100</u>**	**0.235**
SUM + RC^+	0.881	0.060	0.230	0.285	0.098	0.232
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 17: RQ2 results (pairing of the summarization task family with another task family) for Reddit TIFU and the sequential strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
SUM + CLS	**0.881**	0.061	**0.233**	**0.287**	0.096	**0.232**
SUM + CMNS	**0.881**	0.059	0.231	0.284	0.097	0.230
SUM + NLI	**0.881**	**0.062**	**0.233**	**0.287**	**0.098**	0.231
SUM + RC	**0.881**	0.059	0.229	0.286	0.097	0.231
SUM + RC^+	**0.881**	0.057	0.225	0.283	0.096	0.229
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 18: RQ2 results (pairing of the summarization task family with another task family) for Reddit TIFU and the simultaneous strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
SUM + CLS	0.864	0.003	0.077	0.093	0.013	0.081
SUM + CMNS	**0.881**	**0.062**	0.234	0.289	<u>**0.100**</u>	<u>**0.236**</u>
SUM + NLI	**0.881**	0.053	0.223	0.280	0.095	0.225
SUM + RC	**0.881**	**0.062**	**0.234**	<u>**0.290**</u>	<u>**0.100**</u>	0.233
SUM + RC^+	**0.881**	0.061	0.234	0.288	<u>**0.100**</u>	0.233
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 19: RQ2 results (pairing of the summarization task family with another task family) for Reddit TIFU and the continual multi-task learning strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS + CMNS	0.863	0.003	0.078	0.091	0.013	0.081
CLS + NLI	0.864	0.003	0.077	0.093	0.013	0.081
CLS + RC	**0.881**	0.059	0.231	0.288	0.097	0.232
CLS + RC^+	**0.881**	0.059	0.229	0.286	0.097	0.231
CMNS + NLI	**0.881**	0.060	0.231	0.286	0.099	0.231
CMNS + RC	**0.881**	0.059	0.227	0.282	0.096	0.228
CMNS + RC^+	**0.881**	**0.061**	0.232	0.287	0.097	0.231
NLI + RC^+	**0.881**	**0.061**	**0.233**	**0.289**	<u>**0.100**</u>	**0.234**
NLI + RC	**0.881**	0.058	0.231	0.286	0.097	0.231
RC + RC^+	**0.881**	0.058	0.228	0.284	0.096	0.230
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 20: RQ3 results (pairing of two task families excluding the text summarization family) for Reddit TIFU and the sequential strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset, independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS + CMNS	0.863	0.003	0.078	0.091	0.013	0.081
CLS + NLI	0.864	0.003	0.077	0.093	0.013	0.081
CLS + RC	**0.881**	0.059	0.231	0.288	0.097	0.232
CLS + RC^+	**0.881**	0.059	0.229	0.286	0.097	0.231
CMNS + NLI	**0.881**	0.060	0.231	0.286	0.099	0.231
CMNS + RC	**0.881**	0.059	0.227	0.282	0.096	0.228
CMNS + RC^+	**0.881**	**0.061**	0.232	0.287	0.097	0.231
NLI + RC	**0.881**	0.058	0.231	0.286	0.097	0.231
NLI + RC^+	**0.881**	**0.061**	**0.234**	**0.289**	<u>**0.100**</u>	**0.234**
RC + RC^+	**0.881**	0.058	0.228	0.284	0.096	0.223
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 21: RQ3 results (pairing of two task families excluding the text summarization family) for Reddit TIFU and the simultaneous strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS + CMNS	0.853	0.002	0.060	0.095	0.012	0.085
CLS + NLI	0.869	0.000	0.046	0.056	0.007	0.055
CLS + RC	**0.881**	0.060	0.230	0.286	0.099	0.232
CLS + RC+	0.863	0.003	0.082	0.095	0.012	0.085
CMNS + NLI	0.865	0.002	0.081	0.099	0.012	0.089
CMNS + RC	0.864	0.003	0.077	0.093	0.013	0.081
CMNS + RC+	**0.881**	**0.062**	**0.232**	**0.287**	**0.099**	**0.233**
NLI + RC	**0.881**	0.060	0.231	0.287	0.098	0.232
NLI + RC+	**0.881**	0.057	0.227	0.283	0.096	0.229
RC + RC+	**0.881**	0.059	0.228	0.284	0.098	0.230
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 22: RQ3 results (pairing of two task families excluding the text summarization family) for Reddit TIFU and the continual multi-task learning strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
SUM + CLS + CMNS	0.881	0.060	0.228	0.286	0.098	0.232
SUM + CLS + NLI	0.881	0.059	0.231	0.285	0.098	0.231
SUM + CLS + RC	<u>**0.882**</u>	0.060	**0.235**	0.288	0.099	**0.234**
SUM + CLS + RC+	0.881	**0.062**	**0.235**	0.288	<u>**0.100**</u>	0.232
SUM + CMNS + NLI	0.881	0.059	0.230	0.284	0.096	0.229
SUM + CMNS + RC	<u>**0.882**</u>	0.061	0.234	0.288	0.099	0.232
SUM + CMNS + RC+	0.881	**0.062**	0.232	0.287	<u>**0.100**</u>	0.233
SUM + NLI + RC	0.881	0.060	0.229	0.283	0.096	0.230
SUM + NLI + RC+	0.881	0.061	0.234	**0.289**	0.099	**0.234**
SUM + RC + RC+	<u>**0.882**</u>	0.058	0.227	0.284	0.099	0.232
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 23: RQ4 results (pairing of the summarization task family with two other task families) for Reddit TIFU and the sequential strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
SUM + RC+ + CLS	**0.881**	0.061	0.233	0.289	0.099	0.232
SUM + RC+ + CMNS	**0.881**	0.061	0.231	0.286	0.099	0.232
SUM + RC+ + NLI	**0.881**	0.058	0.229	0.285	0.098	0.231
SUM + RC+ + RC	**0.881**	0.059	0.234	0.287	0.097	0.232
SUM + CLS + CMNS	**0.881**	0.057	0.227	0.283	0.096	0.229
SUM + CLS + NLI	**0.881**	0.060	0.231	0.284	0.099	0.229
SUM + CLS + RC	**0.881**	0.058	0.228	0.286	0.098	0.230
SUM + CMNS + NLI	**0.881**	<u>**0.064**</u>	<u>**0.236**</u>	**0.289**	<u>**0.100**</u>	**0.233**
SUM + CMNS + RC	**0.881**	0.061	0.232	0.288	0.099	**0.233**
SUM + NLI + RC	**0.881**	0.061	0.231	0.287	0.098	**0.233**
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 24: RQ4 results (pairing of the summarization task family with two other task families) for Reddit TIFU and the simultaneous strategy. Values in **bold** represent the highest results for a training scheme. <u>Underlined</u> values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
SUM + CLS + CMNS	0.864	0.003	0.077	0.093	0.013	0.081
SUM + CLS + NLI	0.863	0.003	0.082	0.095	0.012	0.085
SUM + CLS + RC	**0.881**	0.058	0.229	**0.285**	0.098	0.231
SUM + CLS + RC+	0.863	0.003	0.082	0.095	0.012	0.085
SUM + CMNS + NLI	**0.881**	**0.059**	0.229	**0.285**	0.098	0.230
SUM + CMNS + RC	**0.881**	**0.059**	**0.230**	**0.285**	**0.099**	**0.232**
SUM + CMNS + RC+	**0.881**	**0.059**	0.228	0.284	0.096	0.229
SUM + NLI + RC	**0.881**	**0.059**	0.228	0.284	0.096	0.230
SUM + NLI + RC+	**0.881**	0.058	0.229	0.284	0.096	0.230
SUM + RC + RC+	**0.881**	**0.059**	0.228	**0.285**	0.097	0.230
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 25: RQ4 results (pairing of the summarization task family with two other task families) for Reddit TIFU and the contniual multi-task learning strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS + CMNS + NLI	0.752	0.000	0.034	0.044	0.000	0.040
CLS + CMNS + RC	**0.881**	**0.062**	**0.235**	0.287	**0.099**	0.231
CLS + CMNS + RC+	**0.881**	**0.062**	0.231	0.286	0.098	0.232
CLS + NLI + RC	**0.881**	0.059	0.233	**0.289**	**0.099**	**0.233**
CLS + NLI + RC+	**0.881**	0.059	0.232	0.286	0.097	0.231
CLS + RC + RC+	0.880	0.060	0.232	0.285	0.098	0.230
CMNS + NLI + RC	0.880	0.059	0.229	0.284	0.095	0.230
CMNS + NLI + RC+	**0.881**	0.059	0.231	0.284	0.096	0.230
CMNS + RC + RC+	**0.881**	0.058	0.232	0.285	0.097	0.230
NLI + RC + RC+	**0.881**	0.058	0.230	0.284	0.097	0.229
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 26: RQ4 results (pairing of three task families excluding the text summarization family) for Reddit TIFU and the sequential strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS + CMNS + NLI	0.746	0.000	0.024	0.028	0.000	0.275
CLS + CMNS + RC	**0.881**	0.060	0.232	0.287	**0.099**	**0.232**
CLS + CMNS + RC+	0.863	0.003	0.082	0.095	0.012	0.085
CLS + NLI + RC	**0.881**	0.059	0.228	0.285	0.098	0.230
CLS + NLI + RC+	**0.881**	0.057	0.225	0.283	0.097	0.231
CLS + RC + RC+	**0.881**	0.058	0.227	0.282	0.097	0.229
CMNS + NLI + RC	0.766	0.000	0.020	0.009	0.000	0.009
CMNS + NLI + RC+	**0.881**	0.058	0.230	0.283	0.097	0.229
CMNS + RC + RC+	**0.881**	**0.061**	**0.234**	**0.288**	0.097	0.231
NLI + RC + RC+	**0.881**	0.059	0.230	0.284	0.096	0.229
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 27: RQ4 results (pairing of three task families excluding the text summarization family) for Reddit TIFU and the simultaneous strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS + CMNS + NLI	0.751	0.000	0.017	0.000	0.000	0.000
CLS + CMNS + RC	0.753	0.000	0.009	0.015	0.000	0.015
CLS + CMNS + RC+	0.861	0.002	0.064	0.057	0.012	0.054
CLS + NLI + RC	0.864	0.003	0.077	0.093	0.013	0.081
CLS + NLI + RC+	0.863	0.003	0.082	0.095	0.012	0.085
CLS + RC + RC+	0.747	0.000	0.025	0.020	0.000	0.020
CMNS + NLI + RC	0.867	0.004	0.105	0.125	0.012	0.101
CMNS + NLI + RC+	**0.881**	0.058	0.228	0.285	0.096	0.230
CMNS + RC + RC+	**0.881**	**0.060**	0.229	0.284	**0.098**	0.230
NLI + RC + RC+	**0.881**	0.059	**0.231**	**0.286**	**0.098**	**0.231**
BART	0.858	0.003	0.087	0.105	0.011	0.090

Table 28: RQ4 results (pairing of three task families excluding the text summarization family) for Reddit TIFU and the continual multi-task learning strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS	0.820	0.018	0.154	0.248	0.048	0.163
CMNS	**0.860**	0.119	0.286	**<u>0.432</u>**	**0.167**	**<u>0.249</u>**
NLI	0.817	0.020	0.168	0.266	0.048	0.169
RC	0.859	0.117	0.282	0.427	0.165	0.247
RC+	0.859	0.117	0.282	0.426	0.164	0.246
SUM	0.859	**0.121**	**0.288**	0.431	**0.167**	**<u>0.249</u>**
BART	0.859	0.116	0.281	0.425	0.163	0.246

Table 29: RQ1 results (single task family) for arXiv and the sequential strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS	**0.860**	**0.120**	0.287	0.430	**0.167**	**0.248**
CMNS	0.806	0.011	0.197	0.215	0.038	0.137
NLI	0.812	0.006	0.111	0.187	0.016	0.123
RC	0.859	0.119	0.284	0.430	0.166	**0.248**
RC+	0.859	**0.120**	**0.289**	**0.431**	**0.167**	**0.248**
SUM	0.859	0.117	0.282	0.429	0.166	**0.248**
BART	0.859	0.116	0.281	0.425	0.163	0.246

Table 30: RQ1 results (single task family) for arXiv and the simultaneous strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS	0.859	**0.119**	**0.286**	0.429	0.166	0.248
CMNS	0.819	0.017	0.163	0.295	0.051	0.171
NLI	0.815	0.018	0.182	0.272	0.044	0.170
RC	0.859	0.117	0.282	0.426	0.164	0.246
RC+	0.817	0.020	0.203	0.249	0.051	0.159
SUM	**0.860**	**0.119**	**0.286**	**0.431**	**0.167**	**<u>0.249</u>**
BART	0.859	0.116	0.281	0.425	0.163	0.246

Table 31: RQ1 results (single task family) for arXiv and the continual multi-task learning strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
ALL	0.859	0.116	0.281	**0.427**	**0.165**	**0.248**
BART	0.859	0.116	0.281	0.425	0.163	0.246

Table 32: RQ1 results (all task families) for arXiv and the sequential strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
ALL	0.859	0.115	0.279	0.425	**0.164**	0.246
BART	0.859	0.116	0.281	0.425	0.163	0.246

Table 33: RQ1 results (all task families) for arXiv and the simultaneous strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
ALL	0.729	0.000	0.008	0.009	0.000	0.009
BART	0.859	0.116	0.281	0.425	0.163	0.246

Table 34: RQ1 results (all task families) for arXiv and the continual multi-task learning strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
SUM + CLS	**0.860**	**0.119**	**0.285**	**0.430**	**0.167**	0.249
SUM + CMNS	0.811	0.016	0.153	0.254	0.046	0.164
SUM + NLI	0.859	0.117	0.282	0.427	0.165	0.247
SUM + RC	0.859	**0.119**	**0.285**	**0.430**	0.166	0.248
SUM + RC+	0.859	0.118	0.284	0.428	0.166	0.247
BART	0.859	0.116	0.281	0.425	0.163	0.246

Table 35: RQ2 results (pairing of the summarization task family with another task family) for arXiv and the sequential strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
SUM + CLS	**0.860**	0.119	0.285	0.429	0.166	0.247
SUM + CMNS	**0.860**	0.119	0.286	**0.432**	**0.167**	**0.249**
SUM + NLI	0.859	**0.120**	**0.287**	0.431	**0.167**	**0.249**
SUM + RC	0.859	0.115	0.280	0.427	0.164	0.247
SUM + RC+	0.859	0.116	0.281	0.427	0.164	0.247
BART	0.859	0.116	0.281	0.425	0.163	0.246

Table 36: RQ2 results (pairing of the summarization task family with another task family) for arXiv and the simultaneous strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
SUM + CLS	0.859	0.117	0.283	0.429	0.165	0.248
SUM + CMNS	**0.860**	**0.120**	**0.288**	**0.432**	**0.167**	**0.249**
SUM + NLI	0.859	0.117	0.282	0.427	0.165	0.247
SUM + RC	0.859	0.118	0.283	0.428	0.166	0.247
SUM + RC+	0.859	0.118	0.284	0.428	0.166	0.247
BART	0.859	0.116	0.281	0.425	0.163	0.246

Table 37: RQ2 results (pairing of the summarization task family with another task family) for arXiv and the continual multi-task learning strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS + CMNS	**0.863**	0.003	0.078	0.091	0.013	0.081
CLS + NLI	0.731	0.000	0.050	0.086	0.000	0.050
CLS + RC	0.859	0.116	**0.287**	0.427	0.165	0.247
CLS + RC+	0.859	0.118	0.284	0.430	**0.167**	**0.248**
CMNS + NLI	0.821	0.010	0.137	0.261	0.045	0.176
CMNS + RC	0.860	0.117	0.283	0.429	0.165	**0.248**
CMNS + RC+	0.859	0.115	0.279	0.426	0.164	0.247
NLI + RC	0.859	0.119	0.285	0.429	0.166	**0.248**
NLI + RC+	0.859	**0.119**	0.286	**0.431**	**0.167**	**0.248**
RC + RC+	0.859	0.116	**0.287**	0.428	0.165	**0.248**
BART	0.859	0.116	0.281	0.425	0.163	0.246

Table 38: RQ3 results (pairing of two task families excluding the text summarization family) for arXiv and the sequential strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS + CMNS	0.704	0.000	0.050	0.076	0.000	0.046
CLS + NLI	0.743	0.000	0.003	0.006	0.000	0.006
CLS + RC	0.859	0.118	0.283	0.428	0.165	0.247
CLS + RC+	0.859	0.120	0.288	**0.432**	0.167	0.248
CMNS + NLI	0.805	0.012	0.212	0.215	0.041	0.134
CMNS + RC	0.859	0.118	0.284	0.428	0.165	0.248
CMNS + RC+	0.859	0.115	0.280	0.426	0.165	0.247
NLI + RC	0.859	0.119	0.285	0.430	0.166	0.248
NLI + RC+	0.859	**0.121**	**0.290**	**0.432**	**0.168**	**0.249**
RC + RC+	0.859	0.116	0.281	0.426	0.164	0.247
BART	0.859	0.116	0.281	0.425	0.163	0.246

Table 39: RQ3 results (pairing of two task families excluding the text summarization family) for arXiv and the simultaneous strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Task Families	BERTScore	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
CLS + CMNS	0.813	0.018	0.162	0.259	0.052	0.176
CLS + NLI	0.859	0.113	0.276	0.422	0.161	0.245
CLS + RC	0.810	0.018	0.181	0.269	0.048	0.168
CLS + RC+	0.860	**0.120**	**0.287**	**0.432**	**0.167**	**0.249**
CMNS + NLI	0.806	0.009	0.118	0.181	0.016	0.117
CMNS + RC	0.812	0.019	0.179	0.282	0.041	0.157
CMNS + RC+	**0.863**	0.003	0.082	0.095	0.117	0.085
NLI + RC	0.860	0.118	0.284	0.429	0.166	0.247
NLI + RC+	0.859	0.117	0.282	0.426	0.164	0.246
RC + RC+	0.859	0.118	0.285	0.429	0.165	0.248
BART	0.859	0.116	0.281	0.425	0.163	0.246

Table 40: RQ3 results (pairing of two task families excluding the text summarization family) for arXiv and the continual multi-task learning strategy. Values in **bold** represent the highest results for a training scheme. Underlined values are the highest results for that dataset independent of training.

Hyper parameter	**Value**
Optimizer	AdamW
Adam-betas	(0.9, 0.999)
Adam-eps	1e-8
LR	5e-05
LR Scheduler	linear decay
Dropout	0.1
Weight Decay	0
Warmup Updates	0

Table 41: Hyperparameters used throughout all pre-finetuning and finetuning experiments.