IRSum: One Model to Rule Summarization and Retrieval

Sotaro Takeshita[1], Simone Paolo Ponzetto[1], Kai Eckert[2]
[1]Data and Web Science Group, University of Mannheim, Germany
[2]Mannheim University of Applied Sciences, Mannheim, Germany
{sotaro.takeshita, ponzetto}@uni-mannheim.de
k.eckert@hs-mannheim.de

Abstract

Applications that store a large number of documents often have summarization and retrieval functionalities to help users digest large amounts of information efficiently. Currently, such systems need to run two task-specific models, for summarization and retrieval, redundantly on the same set of documents. An efficient approach to amend this redundancy would be to reuse hidden representations produced during the summary generation for retrieval. However, our experiment shows that existing models, including recent large language models, do not produce retrieval-friendly embeddings during summarization due to a lack of a contrastive objective during their training. To this end, we introduce a simple, cost-effective training strategy which integrates a contrastive objective into standard summarization training without requiring additional annotations. We empirically show that our model can perform on par or even outperform in some cases compared to the combination of two task-specific models while improving throughput and FLOPs by up to 17% and 20%, respectively.[1]

1 Introduction

An increase in textual information has been observed in various domains, posing challenges in content discovery and driving extensive efforts in the development of summarization and information retrieval systems. The former aims to produce a shorter version of a given document which encapsulates its essential information (Rush et al., 2015; Zhang et al., 2020), and in the context of the latter, a number of text encoders have been introduced which output document embeddings that can match the query embedding to retrieve relevant documents (Zhuang et al., 2023; Ni et al., 2021; Xu et al., 2023). While the output format from each approach differs, i.e., a summarization model generates a text and a text encoder produces a vector, due to the shared motivation, systems with a large number of documents often apply these two models to the same set of documents. For instance, paper-searching platforms apply both summarization and encoder models to their collection of scientific (Kinney et al., 2023; Takeshita et al., 2024b) or news documents (Bambrick et al., 2020). However, with existing methods, such systems need to run two models for each document – one for summarizing and one for encoding. This is an inefficient and expensive process, especially with the current trend of increasing model sizes (Touvron et al., 2023; Jiang et al., 2023). One possible solution for this issue would be a model that generates a summary as well as a text embedding for the retrieval of an input document at the same time. However, regardless of its practical value, there is no work that targets this setup.

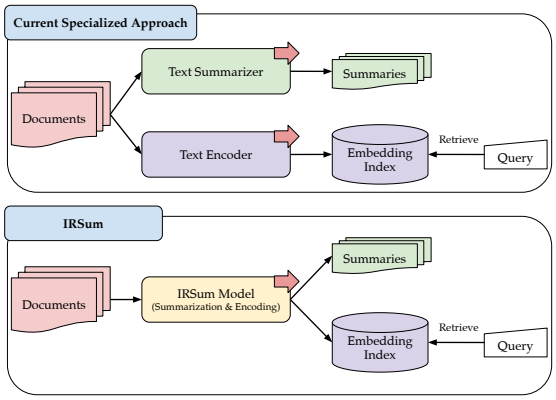

Figure 1: An existing system requires two models to get summary and text embedding, while our single model can produce both in a single forward pass.

To fill this gap, we define a new task in which a single model needs to solve summarization and retrieval **within the same forward pass**, dubbed IRSum. In IRSum, a model must produce hidden representations suitable for retrieval during

[1]https://github.com/sobamchan/irsum

Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM² 2025), pages 262–275

the summary generation, as summarized in Figure 1. In order to evaluate the effectiveness of our approach, we extend three existing summarization datasets to enable retrieval evaluation using the same set of documents. Using these newly constructed datasets, we benchmark a pre-trained language model (PLM), T5, introduced by Raffel et al. (2020), as well as two large language models (LLMs), namely LLaMA 2 7B (Touvron et al., 2023) and Mistral 7B (Jiang et al., 2023). While these models produce high-quality summaries, retrieval performance achieved by the embeddings obtained during the summary generation is well below par with reference baselines, calling for additional learning to unlock the retrieval ability of these models' embeddings.

To this end, we propose a simple multitask training strategy that combines a contrastive objective with a summarization objective. Our method only requires standard summarization datasets for training, and only a small change is needed for its implementation. Our experimental results show that our approach retains both summarization and retrieval abilities close to the combination of two specialized models. Our model can achieve 90% performance for each task while requiring 20% fewer FLOPs and can process 17% more documents per second compared to the existing approach.

Our contributions are as follows. (1) We define a new task, IRSum, that evaluates a model's ability to produce **a summary and embedding for retrieval with only one forward pass**, coupled with extensions of three datasets to achieve its evaluation. (2) We benchmark strong baseline models, including LLM-based summarization models and show that, in contrast to their high-performing summarization ability, **their text embeddings are far from being satisfactory for retrieval**. (3) We propose a simple and efficient multitask training strategy and show **our model achieves comparable performance to the two specialized models with various efficiency improvements**.

2 IRSum

In this section, we first formalize the evaluation of IRSum, then describe how we extend existing summarization datasets for its operationalization, and finally benchmark existing models.

2.1 Task formulation.

IRSum consists of the task to generate a summary and an embedding of a document within one forward pass. The former needs to capture the essential information of the document, while the latter should capture the semantic similarities needed for text retrieval. The evaluation procedure for a model in IRSum is composed of three steps. (1) Inference: the model processes all the test documents and produces summaries and embeddings for each document. (2) Summary evaluation: for each generated summary, we compute ROUGE-2 (Lin, 2004)[2] and G-Eval (Liu et al., 2023)[3]. (3) Retrieval evaluation: by following the recent works on dense retrieval (Khramtsova et al., 2024; Karpukhin et al., 2020), we encode a query using the same model and retrieve the relevant documents using cosine similarity. Then, we use MAP@10 and nDCG@10 to measure the retrieval performance.

2.2 Constructing IRSum datasets.

An essential prerequisite to IRSum is a set of documents with label annotations for both summarization and retrieval. To achieve scalable construction, we draw inspiration from previous works which produce large-scale datasets by exploiting metadata attached to documents. For instance, the MTEB benchmark (Muennighoff et al., 2023a) contains datasets such as SciDocs (Cohan et al., 2020) or CQADupStack (Hoogeveen et al., 2015) which regard titles as queries and the corresponding documents as documents to be retrieved. The same approach can be found in a popular retrieval benchmark, BEIR (Thakur et al., 2021). Other than for benchmarking purposes, works such as those from MacAvaney et al. (2022) or Singh and Singh (2022) take the same approach to achieve a controlled setup for detailed analysis of retrieval models. In this work, by following the aforementioned works, we extend existing summarization datasets by coupling document-summary pairs with titles. One resulting data sample in an extended dataset is a triple composed of a document, summary, and query. As instantiations of our task formulation, we extend three summarization datasets, namely, SciTLDR

[2]We opted for ROUGE-2 over other ROUGE variants due to its highest correlation with humans (Fabbri et al., 2021b). We use py-rouge for its implementation.

[3]We use an open-weight model as its underlying model for reproducible evaluation, namely LLaMA 3. We use the 70B variant for SciTLDR and ACLSum and the 8B model for SQuALITY due to high memory consumption with long inputs.

	SciTLDR		ACLSum		SQuALITY	
Model	**R-2**	**MAP**	**R-2**	**MAP**	**R-2**	**MAP**
ST5$_{\text{BASE/200M}}$	N/A	0.399	N/A	0.427	N/A	0.313
T5$_{\text{BASE/200M}}$	21.47	0.015	16.49	0.039	6.37	0.129
LLaMA-2$_{\text{7B}}$	22.85	0.091	20.85	0.091	8.40	0.127
Mistral$_{\text{7B}}$	23.20	0.008	21.74	0.043	8.18	0.150

Table 1: Performance of fine-tuned T5$_{\text{BASE/200M}}$, LLaMA-2$_{\text{7B}}$ and Mistral$_{\text{7B}}$. The scores of ACLSum are averaged performance over three aspect subsets. We use the contrastively fine-tuned T5 (ST5$_{\text{BASE/200M}}$) as a baseline for retrieval.

(Cachola et al., 2020), ACLSum (Takeshita et al., 2024a), and SQuALITY (Wang et al., 2022) for our experiments. Since the documents in each summarization dataset for the retrieval corpus pool would be too small to simulate a realistic setup. To this end, we add documents in corpora from the same domain for each dataset as distracting samples (§4.1.1 for details).

2.3 Benchmarking of existing models.

As a showcase of the IRSum task, we benchmark our approach with one PLM and two LLMs, namely T5$_{\text{BASE/200M}}$ (Raffel et al., 2020), LLaMA-2$_{\text{7B}}$ (Touvron et al., 2023), and Mistral$_{\text{7B}}$ (Jiang et al., 2023). We evaluate all models after fine-tuning with the corresponding summarization dataset. For document representations, we use the special tokens' representations emitted during the summarization inference. More specifically, we use representations of the first token for T5 (Ni et al., 2021) and the [EOS] token for LLaMA and Mistral (Ma et al., 2023; Wang et al., 2024). The results are shown in Table 1. As a comparison, we also present the results by Sentence-T5, a contrastively trained T5 (base size, 200M parameters, ST5) introduced by Ni et al. (2021). While all models show strong performance in summarization as measured with ROUGE-2, they perform poorly on the retrieval subtask. This is shown by the comparison with ST5$_{\text{BASE/200M}}$, which outperforms LLMs by a large margin while having a much smaller number of parameters. These initial findings provide the motivation for the development of dedicated models for IRSum.

3 Multitask Model for IRSum

Previously, we showed that even LLM-based summarization models fail at the retrieval part of IRSum. Now, we propose a novel multitask training strategy where a model optimizes for summarization and contrastive objectives simultaneously. We design our training strategy following two principles. (1) Only requiring summarization datasets for training: our method does not require any additional annotations other than pairs of source documents and reference summaries from standard summarization datasets. (2) Simple training: our method is a simple add-on to the standard fine-tuning for summarization without complex additional implementation.

3.1 Preliminaries

3.1.1 Summarization training.

Training for summarization use pairs of source documents and target summaries. For both encoder-decoder and decoder-only architectures, a model takes a source document and generates a candidate summary to which a loss is computed using a reference summary. Following is the formal definition of the loss function for encoder-decoder models.

$$L_{sum}^{enc\text{-}dec} = -\sum_{t=1}^{N} \log p_\phi(y_t|\boldsymbol{x}, \boldsymbol{y}_{<t}), \qquad (1)$$

where the model parametrized by ϕ generates a probabilistic distribution of the next token for the summary (y_t), with t being the current generation step. Its generation is conditioned by the source document ($\boldsymbol{x}$) and previously generated summary tokens ($\boldsymbol{y}_{<t}$). On the other hand, the summarization loss for decoder-only models is formulated as,

$$L_{sum} = -\sum_{t=1}^{N} \log p_\phi(y_t|\boldsymbol{y}_{<t}). \qquad (2)$$

The difference from the encoder-decoder (Eq. 1) is that the source document and the previously generated summary tokens are not separately modelled but the latter is a part of the prior, which gets appended as generated.

3.1.2 Contrastive training.

Training for contrastive objectives typically requires pairs of texts that are semantically related to each other. To obtain such data, existing works use entailment pairs from natural language inference datasets (NLI) (Reimers and Gurevych, 2019; Ni et al., 2021; Xu et al., 2023). Negative pairs are often constructed without annotations by pairing sentences randomly within a training mini-batch.

The contrastive objective we use in this work is the following one:

$$L_{cl} = -\log \frac{e^{cossim(\boldsymbol{h}_i, \boldsymbol{h}_i^+)/\tau}}{\sum_{j=1}^{N} e^{cossim(\boldsymbol{h}_i, \boldsymbol{h}_j^+)/\tau}}, \quad (3)$$

where $\boldsymbol{h}_i$ and $\boldsymbol{h}_i^+$ are a pair of embeddings of related texts, and τ is a hyperparameter to control the similarity temperature. Negative pair construction is done in the denominator, where we pair $\boldsymbol{h}_i$ with other embeddings within a batch, of size N. We use cosine similarity for our similarity measurement. Since recent transformer-based models produce embeddings per token, we need to aggregate the token embeddings to form a document representation ($\boldsymbol{h}$ in Eq. 3). Same as §2.3, we use the first and [EOS] tokens' representations respectively for PLMs (Ni et al., 2021) and LLMs (Ma et al., 2023; Wang et al., 2024).

3.2 Multitask training for joint summarization and retrieval

We next describe how we construct pairs of related texts within summarization training loops to seamlessly achieve contrastive learning and then how we combine the summarization and contrastive losses.

3.2.1 Positive pair construction.

To build pairs of texts that are semantically related, we exploit a property of the relationship between source documents and corresponding summaries, that is a summary of a document should entail the information covered in the source document (Falke et al., 2019; Kryscinski et al., 2020). In other words, we can treat document-summary pairs similarly as premise-hypothesis pairs in NLI. This allows us to seamlessly construct labels needed for contrastive loss within summarization training as documents and summaries are already in use in any standard training algorithms.

3.2.2 Multitask task loss.

We combine two losses, namely summarization loss and document-summary contrastive loss, by simply taking a weighted average of two losses, using the balancing hyperparameter λ. Formally as described as $L_{IRSum} = \lambda * L_{sum} + (1 - \lambda) * L_{cl}$, where λ takes a value between 0 to 1, setting λ to 1 would be a standard training for summarization without contrastive objective.

4 Experimental Study

4.1 Setup

4.1.1 Datasets.

We conduct experiments using the IRSum extended versions of three summarization datasets. **SciTLDR** (Cachola et al., 2020) is a single document summarization dataset composed of scientific articles from machine learning conferences and short overview summaries written by the authors and reviewers. We enlarge the retrieval pool by adding 10k papers[4]. **ACLSum** (Takeshita et al., 2024a) is an aspect-based scholarly document summarization dataset where each paper is annotated with three summaries from different perspectives, namely Challenge, Approach, and Outcome. In our experiments, we treat each aspect subset as an individual dataset and report the averaged results. We add the first 10k documents from the training split of Rohatgi (2022) to the retrieval pool. **SQuALITY** (Wang et al., 2022) is a query-focused summarization dataset derived from novels. Each document is coupled with a reference summary with a focus on the corresponding question. We prepend questions before the documents when feeding to models. We add the first 10 documents from the English portion of Project Gutenberg to the retrieval pool[5].

4.1.2 Models.

We use one PLM and two open-weight LLMs and each of the contrastively trained checkpoints for our experiments. **T5** (Raffel et al., 2020) is an encoder-decoder model with 200 million parameters pre-trained for a denoising autoencoding objective. Since its most popular contrastive variant introduced by Ni et al. (2021) only has the encoder without it being followed by a decoder, we fine-tune the original T5 model using the contrastive loss objective proposed by Khosla et al. (2020) on the concatenation of MultiNLI (Williams et al., 2018) and SNLI (Bowman et al., 2015) datasets. We use the premise-hypothesis pairs labelled as entailment as positive pairs and use in-batch negative sampling to construct negative pairs. In the rest of our paper, we refer to this contrastive counterpart we trained as **ST5**. **Mistral** (Jiang et al., 2023) is a decoder-only model with 7 billion pa-

[4] https://huggingface.co/datasets/CShorten/ML-ArXiv-Papers

[5] https://huggingface.co/datasets/manu/project_gutenberg

Model	FT	SciTLDR R-2	SciTLDR GEval	SciTLDR MAP	SciTLDR nDCG	ACLSum R-2	ACLSum GEval	ACLSum MAP	ACLSum nDCG	SQuALITY R-2	SQuALITY GEval	SQuALITY MAP	SQuALITY nDCG
T5	**Specialized**	21.47	3.15	0.399	0.438	16.49	4.31	0.427	0.471	6.37	1.88	0.230	0.313
	IRSum Org	20.29	3.09	0.245	0.271	15.25	4.21	0.015	0.018	5.81	**2.12**	0.022	0.053
	IRSum Cont	20.86	3.09	**0.576**	**0.612**	12.36	4.13	0.377	0.425	4.33	**2.39**	0.083	0.133
	IRSum Merged	20.94	3.12	**0.490**	**0.526**	**16.77**	4.17	0.169	0.187	5.74	**2.15**	0.041	0.081
Mistral	**Specialized**	23.20	1.55	0.229	0.259	21.74	4.50	0.382	0.423	8.18	2.26	0.193	0.295
	IRSum Orig	**23.25**	**2.01**	0.133	0.155	20.22	**4.51**	0.231	0.256	**8.28**	2.03	0.117	0.173
	IRSum Cont	23.07	1.55	**0.418**	**0.458**	21.23	4.25	0.072	0.091	**8.64**	1.96	0.113	0.199
	IRSum Merged	**23.45**	**2.63**	**0.630**	**0.669**	20.96	**4.51**	**0.605**	**0.654**	**8.71**	1.99	**0.270**	**0.321**
LLaMA	**Specialized**	22.85	2.55	0.007	0.008	20.85	4.48	0.000	0.000	8.40	2.48	0.054	0.122
	IRSum Orig	22.80	1.18	**0.017**	**0.021**	20.16	4.45	**0.040**	**0.052**	8.34	1.97	**0.097**	**0.130**
	IRSum Cont	**23.17**	1.54	**0.027**	**0.038**	18.41	4.12	**0.007**	**0.011**	8.06	2.05	**0.100**	**0.152**
	IRSum Merged	**23.14**	1.17	**0.023**	**0.028**	20.31	4.43	**0.024**	**0.030**	8.21	2.11	**0.094**	**0.145**

Table 2: Performance of existing specialized approaches and our multitask models (IRSum). [Orig]inal is a fine-tuned model from the original pre-trained checkpoint, [Cont]rastive is a contrastively-trained version, and Merged is a checkpoint produced by taking an average of summarization and the contrastive models' parameters. Scores are underlined when they achieve 90% of specialized models, **bolded and underlined** when they surpass the specialized counterparts.

rameters. For the contrastively trained version, we use **E5-Mistral** (Wang et al., 2024) where the original model is trained using synthetic data. **LLaMA** (Touvron et al., 2023) is a decoder-only model also with 7 billion parameters. We use **RepLLaMA** (Ma et al., 2023) which is a result of fine-tuning the original LLaMA on the training split of MS MARCO (Nguyen et al., 2016) for its contrastive counterpart. Additionally, we also evaluate merged checkpoints produced by taking an average between summarization and contrastively fine-tuned models (Wortsman et al., 2022).

4.1.3 Training settings.

We perform a grid search using the validation split for all the model training. We test for learning rate $\in$ {1e-05, 3e-05, 5e-05}. For batch size, we tune $\in$ {16, 8, 4} for T5 and ST5, however, due to their large memory consumption, we set the batch size to 4 with the gradient accumulation of 2 and use QLoRA (Dettmers et al., 2024) fro LLMs. We test $\lambda \in \{0.80, 0.85, 0.90\}$ for our multitask training. We use AdamW optimizer (Loshchilov and Hutter, 2019), and train until the validation loss does not increase for three epochs (i.e., early stopping with the patience of 3). For all the combinations of models and datasets, we perform three fine-tunings using different random seeds and report the average performance.

	Relevance	Consistency	Fluency
Agreement	80%	95%	85%
Specialized $\succ$ IRSum	12	1	1
IRSum $\succ$ Specialized	11	0	2
Tie	17	39	37

Table 3: Result of manual quality evaluation. We calculate the number of times a summary from our multitask model (IRSum) is preferred over one from the specialized model and vice versa. Agreement gives how often two annotators gave the same preference for a pair of summaries.

4.2 Results and discussions

4.2.1 Performance.

Table 2 compares our multitask models to the existing pipelines composed of two task-specific models. In most cases, our multitask models perform on par, e.g., achieving more than 90% of, with the specialized pipelines. In particular, the merged checkpoints enjoy our multitask training, outperforming the specialized models on all the tasks and metrics in retrieval tasks. When Mistral is used as an underlying model, the merged variants also outperform in the summarization task on all datasets on at least one of two evaluation metrics. In addition, we conducted a manual evaluation. To this end, two annotators compare summaries of the first 20 documents from SciTLDR's test split generated by Mistral-based multitask and specialized models according to three aspects (Fabbri et al., 2021a).

Model	Storage (↓)	Batch Size (↑)	FLOPs (↓)	TP (↑)
T5	50.0%	1.3%	1.3%	24.7%
Mistral	49.9%	5.0%	20.4%	17.1%
LLaMA	50.0%	12.5%	20.5%	10.9%

Table 4: Efficiency improvements achieved by our multitask models over existing pipelines using specialized Mistral or LLaMA models across storage, batch size, floating point operations per second (FLOPs) and throughput (TP).

The results are shown in Table 3. The high agreement between the two annotators shows the stability of our study, and the high number of tie cases, especially on Consistency and Fluency, exhibit that the two models produce summaries with the same quality on these metrics. While the number of ties is fewer on Relevance, the win rate between the two models is almost 50%, indicating that there is no significant difference. Based on the results from both automatic and manual evaluations, we conclude that our multitask models can achieve performance comparable to that of the specialized models.

4.2.2 Efficiency.

To assess the efficiency of our multitask models, we compare our models and the specialized pipelines from four perspectives. **Storage**: we check how much disk space is used to store all the files required to run both setups. **Batch Size**: because our multitask model requires less memory at inference time, we can process more documents at once by enlarging the batch size. We find this value by gradually increasing batch size for both setups independently until it causes out-of-memory errors. **FLOPs** counts the number of floating point operations during the inference. We use DeepSpeed's Flops Profiler for its implementation (Rasley et al., 2020). **Thoughput (TP)** shows how many documents can be processed within one second. Table 4 shows the results in the relative improvements achieved by our models when compared to the traditional pipelines. As naturally expected, the required storage size is reduced by half with our method. Because our setup is more memory efficient, we achieve loading up to 12% more samples within one batch, as well as with fewer FLOPs, and finally, we achieve up to 17% higher throughput. Together with our performance results from the previous section, we conclude that our approach can substantially improve computational efficiency while retaining models' performance compared to the existing specialized pipelines.

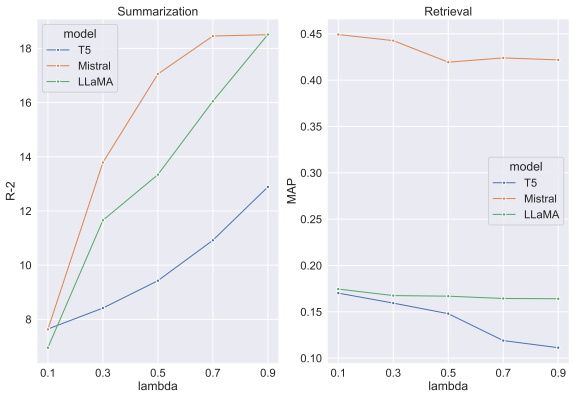

Figure 2: Effect of λ on downstream tasks, summarization (left) and retrieval (right) for different models. The scores are averaged over the three datasets.

4.2.3 λ trade-off.

A hyperparameter in our multitask training, namely λ, balances the summarization and contrastive losses during training. Since the balancing happens on the loss values, whether this hyperparameter indeed behaves as a balancing knob or if there is a trade-off between two tasks at all in downstream performance is not an axiom. To this end, we train models with different lambdas (from 0.1 to 0.9 with a step size of 0.3); a higher lambda means it uses the summarization loss more. In this experiment, we fix the batch size to 16 and 8, respectively, for T5 and Mistral/LLaMA, and the learning rate to 1e-05 for all models. To reduce the computational cost, we do not perform retrieval pool augmentation in this set of experiments. The results are shown in Figure 2, the scores are averaged over three datasets. Summarization abilities by different models increase as the lambda gets higher (on the right in the Figure), however, the sensitivity of retrieval performance to the lambda is much weaker, as the gaps between MAPs when lambda is 0.1 and 0.9 are less than 0.05 for both Mistral and LLaMA.

Model merging for IRSum. Model merging is recently drawing attention as a training-free alternative method to obtain models for fine-tuning (Jin et al., 2023; Don-Yehiya et al., 2023). The objective of our IRSum task is to replace a specialized pipeline with two models with one multitask model where the model merging can provide a cheap option to produce such a multitask model. To this end, we take the simplest model merging which is to

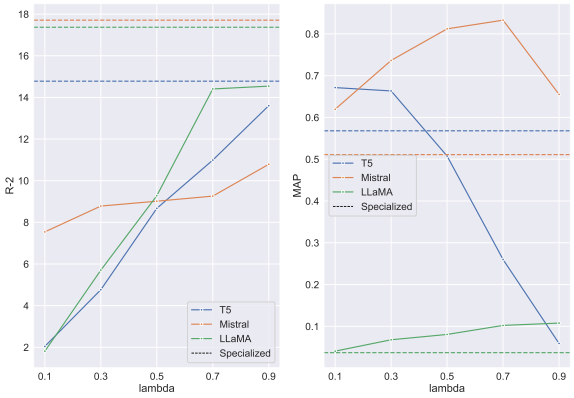

Figure 3: Performance of models obtained by taking weighted (controlled by lambda) averages between summarization and contrastive checkpoints. A higher lambda means that weights from a summarization model are used more, with 0.5 being an exact average of two. Dashed lines are scores achieved by specialized models.

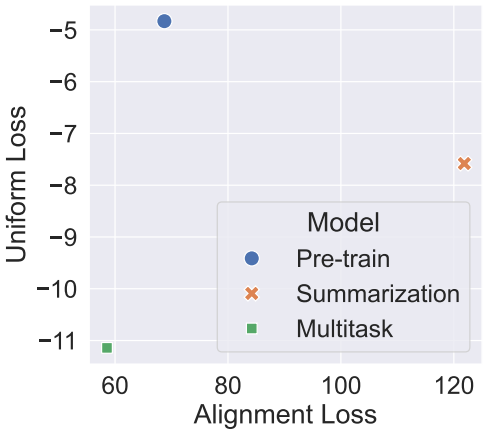

Figure 4: Uniform and alignment losses by only pre-training, standard summarization fine-tuning, and our multitask models. Results are averaged over three datasets.

take a weighted average of two models (Wortsman et al., 2022). Specifically, for each architecture, we merge its contrastive and standard summarization fine-tuned checkpoints. Note that this process does not require any weight updates, hence, this process can be cheaply done without GPUs even for large models. We do not expand retrieval poor for the experiments described in this subsection. The result is shown in Figure 3. Regardless of lambda, the hyperparameter that decides the balance between two models to be merged, all three model architectures degrade summarization performance compared to the original summarization counterparts (dashed lines in the figure) by large margins. Especially, Mistral loses more than 5 ROUGE-2 points even when the lambda is set to 0.9, outperformed by the other two models, including a much smaller, T5. However, for retrieval, surprisingly, all models outperform the retrieval-specialized version with some lambdas. The two LLMs especially outperform the specialized model with all lambdas. However, the positive results on retrieval, due to the lower performance on summarization, we conclude that while model merging can produce well-performing initial checkpoints with fine-tuning (see Table 2), simple merging alone does not result in satisfactory performance.

Representation shift by multitask training. We now perform intrinsic evaluation of embeddings instead of the extrinsic evaluation with downstream tasks to understand the effect of our multitask training in embedding space. To this end, we take two losses, uniform loss and alignment loss, by following the existing works that aim to improve encoder models (Wang and Isola, 2020; Ni et al., 2021). The uniform loss computes how well input embeddings are distributed, which we compute using documents. The alignment loss shows the expected distance between pairs of provided embedding, we use document-query pairs. Lower scores are better for both losses. The result is shown in Figure 4, where we compare how two losses shift when two different fine-tunings are applied to the pre-trained model of T5. One can observe that doing standard summarization fine-tuning improves the embedding space usage indicated by the lower uniform loss than just the pre-trained model; however, the alignment loss increases, meaning that having embeddings close to each other when texts' semantics are related is not a required property for the summarization task. On the contrary, our multitask model improves both losses from the pre-trained model and the standard summarization model. Our models improving uniform loss over the standard summarization model is a possible reason why our models sometimes outperform the specialized model on summarization, as we report in Table 2.

4.2.4 Cross-lingual setup.

Our previous experiments consider monolingual setups where documents, summaries, and queries are all in one language – English. We now test how the specialized approaches and our multitask models perform in a cross-lingual setup where the languages of input and output are different. Specifically, we use the X-SciTLDR dataset (Takeshita

	DE		IT		ZH		JA	
	R-2	MAP	R-2	MAP	R-2	MAP	R-2	MAP
Specialized	9.81	0.273	12.96	0.238	13.56	0.168	5.79	0.242
IRSum **Orig**	9.15	0.210	11.19	0.192	12.94	0.024	5.33	0.074
Cont	9.38	**0.363**	11.49	**0.362**	13.22	**0.212**	5.04	0.170
Mer	9.08	**0.622**	9.45	**0.632**	10.21	**0.622**	**8.68**	**0.622**

Table 5: Performance comparison between the specialized pipeline and our multitask model (IRSum) in cross-lingual setup based on original vs. contrastive vs. merged checkpoints.

et al., 2022), composed of research publications in English and summaries in four different languages. While summaries are already in non-English languages, the queries (i.e., titles for each document) are in English. To achieve a cross-lingual retrieval setup, we translate English titles into four corresponding languages using a distilled version of the NLLB model (Team et al., 2022). We consider Mistral as a base model for this experiment (LLaMA-based models are omitted since RepLLaMA is only trained on English data). For contrastive variants, we use E5-Mistral off-the-shelf since it includes all four languages in its contrastive training stage. The results are shown in Table 5. While our multitask model shows competitive performance to the specialized pipelines, especially its contrastive checkpoint, it successfully achieves 80% in all languages on summarization and outperforms in three languages on retrieval. It does not achieve 80% in Japanese retrieval. This can be due to the fact that the Japanese portion is the smallest in E5-Mistral's contrastive training samples compared to the other languages (Wang et al., 2024). Merged checkpoints show large improvements in retrieval, similar to our monolingual experiments.

5 Related work

5.1 Multitask benchmarks.

Strong interests in models that are capable of solving multiple tasks have driven the development of benchmarks (Wang et al., 2018; Muennighoff et al., 2023b; Gehrmann et al., 2021). However, since the input documents are not shared, they cannot measure the models' ability to make multiple outputs in a single forward pass.

5.2 Multitask models.

In this paper, we take the simplest approach to model multiple losses, that is to take a weighted average between losses, while we achieve satisfactory results with this, there have been several methods with improvements. Mao et al. (2022) propose to use a generalization loss in addition to the standard training loss to improve the balance between tasks. Another work by Chai et al. (2023) introduces a way to resolve the conflicts between tasks. While these papers focus on different instances of the text classification task, they can improve our simple multitask training strategy, which is left for our future work. A few works also investigated multitask training for text summarization (Guo et al., 2018; Magooda et al., 2021; Kirstein et al., 2022). These works report having auxiliary tasks can improve the target summarization performance, however, they do not consider improving on multiple tasks at the same time as we do in this paper.

5.3 Contrastive learning for text generation models.

In addition to applications for encoder-only models (Ni et al., 2021; Wu et al., 2022; Xu et al., 2023), there have been a few works where contrastive learning is applied for text generation models, aiming to improve text generation performance (Su et al., 2022; An et al., 2022). Jain et al. (2023) propose to continuously train decoder-only GPT-2 on a contrastive objective together with the causal language modelling objective. For text summarization, Cao and Wang (2021) propose to use a contrastive loss as an auxiliary loss and show that it can improve models' faithfulness. However, their integration of contrastive learning focuses on the summarization ability of the model while we are interested in giving summarization models a new retrieval ability.

6 Conclusion

In this paper, we first define a new multi-object task setup which asks a model to summarize and encode a document for retrieval within a single forward pass. We extend three existing summarization datasets so that we can use the same set of documents to evaluate on the two tasks. By using them, we find that existing summarization models based on a PLM and recent LLMs cannot achieve satisfactory performance in this setup. Given this result, we propose a new multitask training strategy which cheaply integrates a contrastive objective into the standard summarization training loop and show that our models often achieve performance

comparable to a combination of two specialized models or even sometimes outperform them while being much more computationally efficient.

7 Limitations

Our work has the following limitations. First, while we consider three summarization datasets with different styles, namely single document, aspect-based, and query-focused summarization, however, there are other types of summarization tasks that practically suitable to our multitask task setup, such as multi-document summarization. Second, we use the simplest approach to combine summarization and contrastive losses in our proposed multitask training strategy, there are more complex and recent approaches such as Mao et al. (2022) where they also take generalization loss into account to balance multiple losses. Due to its simplicity our approach does not support how to achieve multitask inference on passage-level which may be suitable for some retrieval setups. We plan to extend our work towards to these two directions in our future projects.

Acknowledgements

The work presented in this paper is funded by the German Research Foundation (DFG) under the VADIS (PO 1900/5-1; EC 477/7-1) and JOIN-T2 (PO 1900/1-2) projects, and also supported by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1597-1 FUGG. We also thank our colleague Daniel Ruffinelli for his comments on a draft of this paper.

References

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. 2022. Cont: Contrastive neural text generation. *Advances in Neural Information Processing Systems*, 35:2197–2210.

Joshua Bambrick, Minjie Xu, Andy Almonte, Igor Malioutov, Guim Perarnau, Vittorio Selo, and Iat Chong Chan. 2020. NSTM: Real-time query-driven news overview composition at Bloomberg. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 350–361, Online. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. TLDR: Extreme Summarization of Scientific Documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online. Association for Computational Linguistics.

Shuyang Cao and Lu Wang. 2021. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Heyan Chai, Jinhao Cui, Ye Wang, Min Zhang, Binxing Fang, and Qing Liao. 2023. Improving gradient trade-offs between tasks in multi-task text classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2565–2579, Toronto, Canada. Association for Computational Linguistics.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Shachar Don-Yehiya, Elad Venezian, Colin Raffel, Noam Slonim, and Leshem Choshen. 2023. ColD fusion: Collaborative descent for distributed multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 788–806, Toronto, Canada. Association for Computational Linguistics.

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

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021b. SummEval: Re-evaluating summarization evaluation. *Trans. Assoc. Comput. Linguist.*, 9:391–409. Publisher: MIT Press - Journals.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language

inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

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, and 37 others. 2021. 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. Association for Computational Linguistics.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 687–697, Melbourne, Australia. Association for Computational Linguistics.

Doris Hoogeveen, Karin M. Verspoor, and Timothy Baldwin. 2015. Cqadupstack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian Document Computing Symposium*, ADCS '15, New York, NY, USA. Association for Computing Machinery.

Nihal Jain, Dejiao Zhang, Wasi Uddin Ahmad, Zijian Wang, Feng Nan, Xiaopeng Li, Ming Tan, Ramesh Nallapati, Baishakhi Ray, Parminder Bhatia, Xiaofei Ma, and Bing Xiang. 2023. ContraCLM: Contrastive learning for causal language model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6436–6459, Toronto, Canada. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Ekaterina Khramtsova, Shengyao Zhuang, Mahsa Baktashmotlagh, and Guido Zuccon. 2024. Leveraging llms for unsupervised dense retriever ranking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1307–1317, New York, NY, USA. Association for Computing Machinery.

Rodney Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, Miles Crawford, Doug Downey, Jason Dunkelberger, Oren Etzioni, Rob Evans, Sergey Feldman, Joseph Gorney, David Graham, Fangzhou Hu, and 29 others. 2023. The Semantic Scholar Open Data Platform. *arXiv preprint*. ArXiv:2301.10140 [cs].

Frederic Thomas Kirstein, Jan Philip Wahle, Terry Ruas, and Bela Gipp. 2022. Analyzing multi-task learning for abstractive text summarization. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 54–77, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

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.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning llama for multi-stage text retrieval. *Preprint*, arXiv:2310.08319.

Sean MacAvaney, Sergey Feldman, Nazli Goharian, Doug Downey, and Arman Cohan. 2022. ABNIRML: Analyzing the behavior of neural IR models. *Transactions of the Association for Computational Linguistics*, 10:224–239.

Ahmed Magooda, Diane Litman, and Mohamed Elaraby. 2021. Exploring multitask learning for low-resource abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1652–1661, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuren Mao, Zekai Wang, Weiwei Liu, Xuemin Lin, and Pengtao Xie. 2022. MetaWeighting: Learning to weight tasks in multi-task learning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3436–3448, Dublin, Ireland. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023a. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023b. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. *arXiv preprint*. ArXiv:2108.08877 [cs].

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. *J. Mach. Learn. Res.*, 21(140):1–67.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Shaurya Rohatgi. 2022. Acl anthology corpus with full text. Github.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Shruti Singh and Mayank Singh. 2022. The inefficiency of language models in scholarly retrieval: An experimental walk-through. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3153–3173, Dublin, Ireland. Association for Computational Linguistics.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561.

Sotaro Takeshita, Tommaso Green, Niklas Friedrich, Kai Eckert, and Simone Paolo Ponzetto. 2022. X-SCITLDR: Cross-Lingual Extreme Summarization of Scholarly Documents. In *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries*, pages 1–12. ArXiv:2205.15051 [cs].

Sotaro Takeshita, Tommaso Green, Ines Reinig, Kai Eckert, and Simone Paolo Ponzetto. 2024a. Aclsum: A new dataset for aspect-based summarization of scientific publications. *arXiv preprint arXiv:2403.05303*.

Sotaro Takeshita, Simone Ponzetto, and Kai Eckert. 2024b. GenGO: ACL paper explorer with semantic features. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 117–126, Bangkok, Thailand. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David

Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint*. ArXiv:2307.09288 [cs].

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. 2022. SQuALITY: Building a Long-Document Summarization Dataset the Hard Way. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1156, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. *Preprint*, arXiv:2401.00368.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and 1 others. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR.

Xing Wu, Chaochen Gao, Zijia Lin, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022. InfoCSE: Information-aggregated contrastive learning of sentence embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3060–3070, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiahao Xu, Wei Shao, Lihui Chen, and Lemao Liu. 2023. SimCSE++: Improving contrastive learning for sentence embeddings from two perspectives. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12028–12040, Singapore. 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. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. Rankt5: Fine-tuning t5 for text ranking with ranking losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2308–2313, New York, NY, USA. Association for Computing Machinery.

A Appendix

Model	Licence	URL
T5$_{\text{BASE}}$	Apache 2.0	https://huggingface.co/t5-base
Mistral$_{\text{7B}}$	Apache 2.0	https://huggingface.co/mistralai/Mistral-7B-v0.1
Llama 2$_{\text{7B}}$	LLAMA 2 License	https://huggingface.co/meta-llama/Llama-2-7b-hf
E5-Mistral$_{\text{7B}}$	MIT	https://huggingface.co/intfloat/e5-mistral-7b-instruct
RepLLaMA$_{\text{7B}}$	LLAMA 2 License	https://huggingface.co/castorini/repllama-v1.1-mrl-7b-lora-passage
mT5-base$_{\text{580M}}$	Apache 2.0	https://huggingface.co/google/mt5-base
NLLB Distilled$_{\text{600M}}$	CC by NC 4.0	https://huggingface.co/facebook/nllb-200-distilled-600M
SciTLDR	Apache 2.0	https://huggingface.co/datasets/allenai/scitldr
ACLSum	MIT	https://huggingface.co/datasets/sobamchan/aclsum
X-SciTLDR	MIT	https://huggingface.co/datasets/umanlp/xscitldr
SQuALITY	Apache 2.0	https://huggingface.co/datasets/pszemraj/SQuALITY-v1.3

Table 6: A list of datasets and models used in our study with external URLs.