IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 January 2024, accepted 10 March 2024, date of publication 18 March 2024, date of current version 4 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3379139

RESEARCH ARTICLE

Investigating the Pre-Training Bias in Low-Resource Abstractive Summarization

DANIIL CHERNYSHEV[1] AND BORIS DOBROV[1,2]
[1]Research Computing Center, Lomonosov Moscow State University, 119991 Moscow, Russia
[2]ISP RAS Research Center for Trusted Artificial Intelligence, 109004 Moscow, Russia

Corresponding author: Daniil Chernyshev (chdanorbis@yandex.ru)

The work of Daniil Chernyshev was supported by the Non-Commercial Foundation for Support of Science and Education "INTELLECT." The work of Boris Dobrov was supported by a grant for research centers in the field of artificial intelligence, provided by the Analytical Center for the Government of the Russian Federation, under Agreement 000000D730321P5Q0002; and in part by the Ivannikov Institute for System Programming of the Russian Academy of Sciences, in 2021, under Agreement 70-2021-00142.

ABSTRACT Recent advances in low-resource abstractive summarization were largely made through the adoption of specialized pre-training, pseudo-summarization, that integrates the content selection knowledge through various centrality-based sentence recovery tasks. However, despite the substantial results, there are several cases where the predecessor general-purpose pre-trained language model BART outperforms the summarization-specialized counterparts in both few-shot and fine-tuned scenarios. In this work, we investigate these performance irregularities and shed some light on the effect of pseudo-summarization pre-training in low-resource settings. We benchmarked five pre-trained abstractive summarization models on five datasets of diverse domains and analyzed their behavior in terms of extractive intuition and attention patterns. Despite that all models exhibit extractive behavior, some lack the prediction confidence to copy longer text fragments and have a misaligned attention distribution with the structure of the real-world texts. The latter happens to be the major factor of underperformance in fiction, news, and scientific articles domains as the better initial attention alignment of BART leads to the best benchmark results in all few-shot settings. A further examination reveals that BART summarization capabilities are the side-effect of the combination of sentence permutation task and specificities of the pre-training dataset. Based on the discovery we introduce Pegasus-SP, an improved pre-trained abstractive summarization model that unifies pseudo-summarization with sentence permutation. The new model outperforms the existing counterparts in low-resource settings and demonstrates superior adaptability. Additionally, we show that all pre-trained summarization models benefit from data-wise attention correction, achieving up to 10% relative ROUGE improvement on model-data pairs with the largest distribution discrepancies.

INDEX TERMS Abstractive summarization, attention mechanism, low-resource text processing, pre-trained language models, model probing.

I. INTRODUCTION

Automatic summarization as an area of natural language processing studies text processing methods aimed at the extraction of the most important information from the source document. There are two approaches: extractive and abstractive. Extractive methods leverage existing text fragments and rely on various heuristics to combine them. Abstractive methods improve on the extractive by employing additional language resources to paraphrase and fuse the salient fragments, thus producing the most concise summary. While extractive approaches have been developing for several decades, abstractive approaches saw advancements only with the emergence of powerful neural language models.

Initial neural solutions, based on the original sequence-to-sequence, showed promising results in various summarization tasks [35]. These approaches were refined [29], [36] by

The associate editor coordinating the review of this manuscript and approving it for publication was Jihad Aljaam.

integrating the attention mechanism [2] which allowed the abstractive summarization models to surpass the extractive approaches in terms of coherence and relevance. Human-like quality abstractive summarization became possible with the introduction of pre-trained Transformer-based [39] encoder-only language models such as BERT [6] that provide global contextual knowledge through the self-attention mechanism. BERTSum [22] was one of the first models to employ BERT encoder for abstractive summarization and demonstrated substantial improvement over previous approaches. Later it was shown [20] that the performance could be further improved by jointly pre-training the full sequence-to-sequence transformer model on language modeling tasks.

The recent development of abstractive summarization models [13], [31], [42], [45] revolved around task-specific pre-training to alleviate the abstractive summarization in low-resource settings. While these summarization-focused language models proved to be efficient in zero-shot and few-shot tuning settings, they showed the same performance margin after full fine-tuning despite the different pre-training tasks. Moreover, several works [12], [18], [23] indicated that those models produce inferior results compared to general-purpose pre-trained generative language models such as BART even in low-resource settings. Amid the rise of universal NLP problem-solving models (e.g. ChatGPT or GPT-4 [27]) capable of producing summaries of quality comparable to exclusively fine-tuned models [40], [46] these observations call into question the reasonableness of task-specific pre-training in abstractive summarization.

In this work, we investigate the causes of substantial performance differences between summarization-specialized pre-trained models and general-purpose pre-trained generative language models in low-resource tuning scenarios. Our key contributions are:

1) We benchmark pre-trained Transformer-based encoder-decoder language models that previously achieved state-of-the-art results in low-resource abstractive summarization on five summarization datasets of diverse domains (see Table 1 for the summary of our results). The evaluation confirms past observations, demonstrating the efficiency of BART model;
2) We provide an analysis of model extractive intuition and attention distribution pattern dynamics and show their connection to performance irregularities in low-resource settings. We conclude that the underperformance of pre-trained summarization models is mainly caused by mismatched salient position assumptions of pre-training tasks;
3) We reveal that BART summarization capabilities come from the combination of sentence permutation task and specificities in pre-training data. Based on the discovery we introduce Pegasus-SP, an improved variant of Pegasus with superior adaptability and low-resource performance.

TABLE 1. Average language model ranking in low-resource summarization and their best domains.

Model	Model size	Avg. rank 0-shot	Avg. rank 1000-shot	Best domain
BART	406M	2.5	**1.7**	Patents
Pegasus	570M	4.0	3.2	Instructions
LongT5	770M	5.3	5.2	Instructions
PRIMERA	447M	2.3	5.2	Science
Centrum	**152M**	5.3	4.2	Literature
Pegasus-SP (ours)	570M	**1.5**	**1.7**	News

4) We demonstrate the importance of attention and salient sentence distribution alignment by testing three attention shifting methods. By alleviating the pre-training attention bias on model-data pairs with the largest distribution discrepancies we achieve up to 10% ROUGE relative improvement over the original model.

II. RELATED WORK

Behavioral patterns of abstractive summarization systems has been actively studied with the introduction of powerful universal models [22], [36]. However, only a few works explored the intrinsic properties of summarization models, due to the substantial computation requirements of probing methods which can scale quadratically with the length of the input sequence. Xu et al. [44] studied the decoder token-wise attention and decision uncertainty during the generation process of fine-tuned BART [20] and Pegasus [45] models and revealed a positive correlation between the level of summary abstractiveness and model entropy. Wilber et al. [41] conducted a similar analysis of pointer-generator network [36] in the context of lexical and syntactic features to understand the factors behind paraphrasing capabilities. Xu et al. [43] evaluated the ability of the explanation methods to detect the most relevant input fragments in abstractive summarization models and showed that over 20% of generation decisions ignore contextual information from the source document. Zou et al. [47] explored the omission problem in dialogue summarization in pre-trained language models and revealed that in at least 70% of cases the models make omission errors. DeYoung et al. [7] studied the specifics of synthesis performed by multi-document abstractive summarization models and demonstrated that existing pre-trained language models are both oversensitive to input document-wise permutations and insensitive to input composition changes (e.g., positive-negative review ratio). Gonzalez et al. [11] examined entity aggregation in abstractive summarization and showed that the pre-trained language models are unable to replicate this type of paraphrasing without copying directly from the source document. Otmakhova et al. [28] tested the ability of pre-trained language models for multi-document abstractive summarization to follow the evidence in the medical documents using prediction difference technique [21] and revealed that models tend to ignore negations or overall statement polarities and prone to cite neutral sentences that don't make any claims. Chen et al. [48] benchmarked different training strategies for low-resource summarization

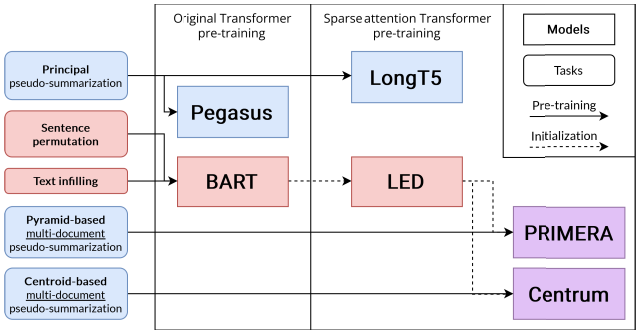

FIGURE 1. Relations between pre-trained abstractive summarization models.

on 8 datasets and showed that additional multi-task pre-training with task-specific prefixes substantially boosts the adaptability in few-shot tuning scenarios.

III. PROBING THE PRE-TRAINED SUMMARIZATION MODELS IN A LOW-RESOURCE SETTING

The goal of our experiments is to study available pre-trained abstractive summarization models in terms of low-resource efficiency. We consider summarization-specialized pre-trained models that achieved state-of-the-art results on standard evaluation datasets in low-resource settings (single- or multi- document summarization) and have publicly available weights (Section III-A). To test the models, we first benchmark them on five popular datasets of diverse domains (Section III-B) and then evaluate their content selection capabilities

A. MODELS

Figure 1 shows the relationship between considered pre-trained summarization models. Large language models like ChatGPT or GPT-4 [27] are out of scope since the results of their summarization are heavily dependent on the user's prompt [40] and thus can't be objective without an exhaustive prompt search. Further we describe the features of each model.

BART [20]. One of the first Transformer encoder-decoder pre-trained language models designed for text generation tasks. The model follows the original symmetrical architecture [39] with 12 layers in each encoder and decoder. To inject general language knowledge the model is pre-trained on two text denoising tasks: text infilling (recovering masked token sequences) and sentence permutation (recovering the original sentence order). Despite the lack of summarization-specialized pre-training the model achieved state-of-the-art results in low-resource settings [18], [23], [42].

Pegasus [45]. The idea of pseudo-summarization pre-training was introduced with Pegasus Transformer language model. Having the same architecture as BART, Pegasus pre-trains by recovering the most salient sentence subset (pseudo-summary) using the set remainder as the input text. The salient sentences are determined using the Principal strategy, which selects the sentences with the highest overlap (ROUGE-1 F1 scores) with the rest of the text.

TABLE 2. Dataset statistics.

Domain	Dataset	# examples in Train / Val / Test	# words Source	Target
News	CNN / DM	287K / 13K / 11K	698.6	49.5
Science	ArXiv	203K / 6K / 6K	5179.2	257.4
Instructions	WikiHow	160K / 6K / 6K	579.8	62.1
Patents	Bigpatent	1207K / 67K / 67K	3572.8	116.5
Literature	Booksum - chapter - paragraph	 10K / 1K / 1K 115K / 15K / 15K	 5101.9 159.6	 505.4 40.6

PRIMERA [42]. Pegasus has two issues: low input length limit (1024 tokens) and sub-optimality of principal strategy in multi-document summarization. PRIMERA addressed the first issue by employing LED [3] encoder-decoder model, a modification of BART with a sparse attention mechanism that allows to extend the input context up to 16384 tokens. LED attention mechanism has two separate parts: local diluted sliding window attention and global full token-wise attention. The model is pre-trained on document clusters using a Pyramid-based Principal strategy that prioritizes the sentences with the most mentioned entities. In addition, to distinguish different documents in the input sequences the model utilizes a global attention layer to aggregate the document-wise information into special <doc-sep> token embeddings.

LongT5 [13]. Similarly to PRIMERA, LongT5 addresses the issue of low context length limit in Pegasus model. The model is based on T5.1 [32] pre-trained Transformer language model but uses sparse attention. LongT5 attention is an improvement of local sliding window attention that additionally accumulates the weights within each attention window into transient global meta-tokens. LongT5 is pre-trained on pseudo-summarization with the original Principal strategy.

Centrum [31]. Centrum is a sibling of PRIMERA model trained on an alternative version of the pseudo-summarization task. The authors of the model argued that the named entity pyramid is insufficient for reducing the irrelevance of the pseudo-summary and frequently results in noisy examples that lead to suboptimal performance. To tackle the issue, Centrum uses the document closest to the cluster centroid as the reference summary and the remaining documents as the source.

B. DATASETS

The statistical characteristics of considered datasets are presented in Table 2. To ensure comparability of low-resource benchmark results for each domain we chose datasets that are publicly available or fully reproducible and were most frequently mentioned in works on low-resource single-document abstractive summarization.

CNN / Daily Mail [26]. The original CNN / Daily Mail dataset was proposed for question answering task and was later repurposed for abstractive news summarization thanks to the key point format of the queries. The dataset serves as the standard benchmark for summarization models.

ArXiv [4]. One of the first large-scale long document summarization datasets for the scientific domain. Arxiv leverages science articles from arXiv.org, utilizing the article abstract as the target summary and the rest of the text as the source input.

WikiHow [16]. A large-scale dataset of instructions from the online WikiHow.com website. Each instruction article is presented in the form of multiple paragraphs, each describing the details of the next step. The summaries were obtained by concatenating the titles (highlights) of each instruction step paragraph.

Bigpatent [37]. A collection of United States patent documents across nine different technological areas with over 1 million document-summary pairs scraped from Google Patents Public Dataset. Utilizes an abstract section as the summarization target.

Booksum [18]. A collection of datasets for long-form narrative summarization of texts from the literature domain. Booksum contains human-written summaries on three levels of granularity: paragraph, chapter, and book. Due to input-length limitations of summarization models, in this work, we consider only paragraph and chapter levels.

C. PROBING METHODS

Content filtering in abstractive summarization models is performed in two stages: in the encoder by introducing a saliency signal in final document embeddings and decoder when calculating input content weighting with the cross-attention mechanism. As the saliency signals mixed in encoder embeddings are designed to be decoded only by the decoder of the corresponding configuration the reliable way to estimate their quality is to alleviate exposure bias [34] that affects the decoder's decision-making process and build explicit input-prediction mapping by forcing summarization model's decoder to output the existing fragments of the input document. In particular, we restrict the decoder to output sentence subset, thus reducing the task to sentence ranking. To evaluate attention-based content weighting we use ALTI method [8] which is a Transformer-specialized counterpart of Integrated gradients [38] approach previously shown to produce good explanations for sentence attribution [43].

1) FORCED EXTRACTIVE SETTING

The main difference between abstractive and extractive summarization strategies is the use of paraphrasing tools. Since copying text fragments is easier than generating a novel phrase, most abstractive summarization models exhibit a highly extractive behavior [19]. While this trait is one of the main subjects of criticism [17] the majority of human-written summaries have a low level of abstractiveness as well [10].

If the language model is prone to extractive strategies, it can reliably act as a sentence saliency estimator. On the other hand, if the model tends to generate more abstractive summaries it would underestimate the saliency of sentences with excessive details. To get an estimation we can use the sequence log probability:

$$SeqProb(s, X, \Theta) = \sum_{j=1}^{|s|} \log P(w_j | w_{<j}, X, \Theta) \quad (1)$$

where X is the source document, s is the target sentence,w_j is jth token of s, Θ are model parameters. Given estimations we can find the best extractive prediction by solving the Oracle problem using the greedy algorithm:

$$ApproxOracle(X, \Theta) = arg \max_{x_i \subset X} SeqProb(x_i, X, \Theta) \quad (2)$$

where x_i is a subset of sentences from document X. The upper bound of the solution (**Reference /** Extractive Oracle) is commonly obtained [22] using token overlap metrics against reference summary y such as ROUGE-2:

$$Oracle(X, y) = arg \max_{x_i \subset X} \text{ROUGE-2}(x_i, y) \quad (3)$$

The goal of this test is to evaluate the model's ability to efficiently use the copying mechanism and the stability of the unconstrained generation process (i.e. severity of exposure bias).

2) EXPLORING ATTENTION PATTERNS

Many works exploring the behavior of language models employed model agnostic methods, among which gradient-based attribution [38] and prediction difference [21] showed the best results [14]. The main drawback is the high computational complexity of the methods: prediction difference requires an individual forward pass for each input token (to test the difference after token removal) and gradient requires a backward pass for each output token (to take into account generation timestep difference). A more efficient alternative is to use a model-specific explanation approach such as attention weights.

The attention mechanism plays a major role in the decision-making process of Transformer language models as it controls the attribution of the individual tokens. However, the attention weights of individual layers and heads can't produce faithful explanations since they are not robust to architectural changes [30]. Attention aggregation techniques such as attention rollout [1] are more reliable as they take into account the full information flow. In this work we use the most advanced version of the method, ALTI [8], which outperforms model agnostic methods in terms of alignment error rate [9] and was shown to produce faithful explanations for model hallucinations [5].

Attention rollout [1] simulates the information propagation in Transformer network by viewing the process as a sequential application of attention weight matrices $\bar{A}^l$ to input embedding x^{enc}, thus simplifying the model to the composition of feed-forward layers. For Transformer encoder of L_{enc} layers the scores are obtained as:

$$R = \prod_{l=1}^{L_{\text{enc}}} \bar{A}^l \quad (4)$$

The main issue is that in Transformer's multi-head attention there are H heads and each produces its own local attention weights and the original attention rollout method [1] assumes the equal contribution of each head (i.e. $\bar{A}^l = \frac{1}{|H|}\sum_{h=1}^{H} A_h^l$), however, in Transformer [39] those weights are combined dynamically with W_O projection matrix. ALTI modification addresses the issue by substituting averaged attention weights with exact attention block input-output contributions:

$$(C(x^l, r))_{ij} = \frac{\max\left(0, \|\tilde{y}_i\|_1 - \|\tilde{y}_i - T_i(x^l, j, r)\|_1\right)}{\sum_{k=1}^{n} \max\left(0, \|\tilde{y}_i\|_1 - \|\tilde{y}_i - T_i(x^l, k, r)\|_1\right)} \tag{5}$$

$$\tilde{y}_i = \sum_{j=1}^{n} T_i(x^l, j, r) + \epsilon \tag{6}$$

$$T_i(x^l, j, r) = LN(F_i(x_j) + \mathbb{1}_{j=i} \cdot r) \tag{7}$$

$$F_i(x_j) = \sum_{h=1}^{H} W_O^h a_{ij}^h W_V^h x_j \tag{8}$$

where x_j^l is jth input component of attention block in layer l, $\tilde{y}_i$ is ith output component of attention block, r is residual component (for self-attention $r = x_i^l$, for cross-attention r is the output of preceding self-attention block), $a_{ij}^h \in A_h^l$ is component of head attention weight matrix, LN is component-wise layer norm, ϵ is the sum of attention block bias terms, W_V is attention value transformation matrix. Since Transformer decoder differs from the encoder only by an additional cross-attention layer, which just uses encoder embeddings for attention value calculations (in that case $x^l = \text{Encoder}(x^{\text{enc}})$), equation (5) can be applied to each attention block individually (for simplicity we denote $C_*^l = C(x_*^l, r_*)$). Total encoder-decoder Transformer input contributions $R_{\text{enc-dec}}^l$ for layer $1 \le l \le L_{\text{dec}}$ are obtained as the sum of local self-attention $C_{\text{dec-self}}^l$ and a product of total encoder-only $R_{\text{enc}}^{L_{\text{enc}}}$ and layer-level cross-attention $C_{\text{dec-cross}}^l$ contributions:

$$R_{\text{enc-dec}}^l = C_{\text{dec-self}}^l \cdot R_{\text{enc-dec}}^{l-1} + C_{\text{dec-cross}}^l \cdot R_{\text{enc}}^{L_{\text{enc}}} \tag{9}$$

$$R_{\text{enc-dec}}^1 = C_{\text{dec-cross}}^1 \cdot R_{\text{enc}}^{L_{\text{enc}}} \tag{10}$$

$$R_{\text{enc}}^{L_{\text{enc}}} = \prod_{l=1}^{L_{\text{enc}}} C_{\text{enc}}^l \tag{11}$$

The final rollout scores R of ALTI method are given by total contributions of the last decoder layer $R_{\text{enc-dec}}^{L_{\text{dec}}}$.

ALTI contribution scores can be used without further refinement, however in conditions of high input length variance (for instance news article length ranges from 50 to 2000 words) the token-wise contribution statistics may be misleading without normalization. To alleviate the effect of length disparity and compare the attribution results with Extractive Oracle reference labels we aggregate the contributions on sentence level.

Given ALTI prediction-input total contribution matrix R for document tokens X and generated tokens G we can derive sentence-wise contribution or sentence relevance as the following:

$$SentRel(s, X, G) = \frac{1}{|s|} \sum_{w_j \in s} \sum_{i=0}^{|G|} R_{ij} \tag{12}$$

where $s \subset X$ is the sentence of document X, $w_j \in s$ is a token from sentence s. To determine the set of sentences that reflect the model's extractive intuition we filter out by the upper quartile Q_3 of $SentRel$ score:

$$SalSet(X, G) = \{s_k \subset X | SentRel \ge Q_3(SentRel(*, X, G))\} \tag{13}$$

Comparing the set with the Extractive Oracle we measure the relevance of the most attended sentences in terms of reader's (dataset) expectation:

$$RELAE_{prec}(X, G, y) = \frac{|SalSet(X, G) \cap Oracle(X, y)|}{|SalSet(X, G)|} \tag{14}$$

$$RELAE_{rec}(X, G, y) = \frac{|SalSet(X, G) \cap Oracle(X, y)|}{|Oracle(X, y)|} \tag{15}$$

D. IMPLEMENTATION DETAILS

For few-shot tuning we follow Xiao et al. settings [42]: Adam optimizer with linear scheduled learning rate 3e-5 and warmup ratio 0.1 of total steps, batch size 10, cross-entropy loss with label smoothing, number of maximum training steps is max (200, $total\_examples \cdot 10$), validation every 5 epochs. We select the best models according to ROUGE score of summaries generated with a greedy sampling strategy on the validation set. Since optimal generation configuration depends on both model and dataset to study the pure effect of the tuning procedure we generate summaries for the test set with greedy sampling which produces the summaries of the lower quality boundary. We use model implementations from HuggingFace Transformers library[1] and initialize from large-variant checkpoints ($\ge$ 400 mil parameters), except for Centrum which is based on LED-base version ($\sim$150 mil parameters). For a fair comparison, we limit the input length of all models to 1024 tokens (BART and Pegasus input limit). We sample few-shot training examples from the beginning of the training part of each dataset. We evaluate the quality of summaries with ROUGE-{1, 2, L} F1 scores. To calculate average ALTI scores we use the first 1000 examples from test sets (minimal test size among datasets). To apply the method to sparse-attention architectures we recover the dense representation by combining the global and local components.

E. BENCHMARK RESULTS

The results of standard summarization evaluation for the few-shot tuned model are reported in Table 3. Despite the multi-document specialization PRIMERA outperforms single-document counterparts in zero-shot setting and

[1] https://huggingface.co/docs/transformers

TABLE 3. Mean ± STD of model ROUGE scores across six datasets.

#-shot	Model	ROUGE-1	ROUGE-2	ROUGE-L
0	BART	29.43 ± 6.83	8.24 ± 4.79	17.10 ± 4.04
	Centrum	22.57 ± 7.90	5.85 ± 4.56	14.75 ± 4.76
	LongT5	24.49 ± 5.55	6.29 ± 3.55	15.73 ± 3.76
	Pegasus	26.07 ± 6.33	6.66 ± 3.93	16.50 ± 4.00
	PRIMERA	29.65 ± 6.56	8.19 ± 4.47	17.22 ± 3.77
10	BART	31.19 ± 8.34	8.81 ± 4.45	18.48 ± 4.49
	Centrum	27.40 ± 7.91	7.02 ± 3.86	16.72 ± 3.96
	LongT5	25.24 ± 6.16	6.66 ± 3.87	16.58 ± 4.34
	Pegasus	28.35 ± 7.15	7.45 ± 4.08	18.19 ± 4.32
	PRIMERA	28.87 ± 10.48	8.07 ± 5.19	17.59 ± 5.28
100	BART	34.37 ± 7.79	10.29 ± 4.98	20.58 ± 4.69
	Centrum	30.42 ± 7.20	8.15 ± 4.19	18.46 ± 4.05
	LongT5	29.57 ± 8.24	9.25 ± 5.32	20.27 ± 6.24
	Pegasus	32.55 ± 8.05	9.72 ± 4.81	20.78 ± 5.18
	PRIMERA	32.11 ± 7.94	9.21 ± 4.69	18.92 ± 4.17
1000	BART	35.52 ± 8.09	11.20 ± 5.12	21.93 ± 4.87
	Centrum	32.87 ± 8.62	9.88 ± 4.84	20.29 ± 4.81
	LongT5	33.20 ± 7.91	10.74 ± 5.18	22.02 ± 5.52
	Pegasus	34.12 ± 8.13	10.69 ± 5.07	22.00 ± 5.34
	PRIMERA	32.87 ± 8.62	9.87 ± 4.83	20.29 ± 4.81

BART has the same performance margin while LongT5's and Pegasus's lower performance corroborates previous benchmarks [18], [42].

Few-shot tuning even on 10 examples has a significant effect on summarization quality. Contrary to the trend, the performance of PRIMERA degrades as the model attempts to adapt to a single document setting. The Centrum doesn't share the pattern since the lower number of trainable parameters limits the ability to memorize the pre-training data and encourages exploiting more abstract patterns. LongT5 struggles to improve which may be attributed to a shift in input processing as we force the model to utilize only the 1024 token context window.

Providing 100 examples proves to be sufficient for consistent performance gains. Pegasus is closing in on BART and PRIMERA ranking third while the gap between lower-end models is negligible. At 1000 examples the models seem to approach the optimum as they demonstrate similar quality. However, BART still leads by a significant margin.

Data-wise ROUGE-1 + ROUGE-2 scores (**ROUGE 1+2**) are shown in Figure 2. As can be seen in most cases the pre-trained models exhibit almost linear quality scaling with training examples. The performance anomalies happening at 10 example tuning are linked to specific datasets and domains. The examples from CNN / Daily Mail are likely to lie at the extremes of test distribution as all models experience a drop in performance. However, PRIMERA also happens to experience performance instabilities on Booksum dataset on both summarization levels and hardly improves with additional examples. Centrum on the other hand manages to outperform larger single-document counterparts on ArXiv and Booksum, despite lower comprehension power, indicating the suboptimality of Principal pseudo-summarization pre-training. Booksum generally has the lower ROUGE scores, especially the paragraph variant. This can be explained by an unfamiliar domain that is poorly represented in the pre-training collections as well as the highest level of abstractiveness of reference summaries [18].

TABLE 4. Model performance in Extractive Oracle approximation task.

Oracle Scorer	ROUGE 1+2	Decoder Copy Rate	
		0-shot	1000-shot
Reference [22]	57.32	-	-
BART	26.67	77.26	26.83
Centrum	32.21	51.08	45.74
LongT5	36.92	94.93	32.21
Pegasus	37.14	80.41	27.39
PRIMERA	35.74	79.88	45.96

TABLE 5. Relevant attention evaluation results.

# Examples	Model	RELAE		
		Recall	Precision	F1
0	BART	35.29	27.91	28.14
	Centrum	35.10	27.26	27.63
	Pegasus	28.67	22.90	23.04
	PRIMERA	33.32	25.92	26.30
1000	BART	45.66	34.26	35.49
	Centrum	40.65	31.21	31.92
	Pegasus	43.51	33.16	34.26
	PRIMERA	35.30	27.42	27.85

At 1000 examples the models converge to the similar scores and visually group by architectural traits, specifically by attention density. We presume that the global attention mechanism isn't enough to compensate for the information loss outside the local sliding window and is inefficient for shorter documents. Despite that, LongT5 follows the same score path as Pegasus implying that the pre-training strategy plays a crucial role in abstractive summarization model efficiency. BART demonstrates a consistent superiority throughout low resource evaluation even at 1000 examples. Considering that we are evaluating with lower-bound generation strategy (greedy) this fact raises doubts about the efficiency of summarization-specialized pre-training.

F. PROBING RESULTS

The evaluation of model extractive intuition is reported in Table 4. In addition to ROUGE scores, we report an average percentage of copied fragments (3 or more consequent words) in summaries generated in abstractive mode. The ranking of models is practically the inverse of the abstractive summarization benchmark. BART and PRIMERA perform worse, and the former has almost a 13 loss in absolute quality. At the same time Pegasus, LongT5, and Centrum perform substantially better, with up to 6 ROUGE 1+2 improvement. Taking into account that LongT5 has the highest extractiveness of generated abstractive summaries we can conclude that the model is highly uncertain about its zero-shot predictions and can't reliably copy long text fragments such as sentences. This is likely attributed to Principal pseudo-summarization pre-training as Pegasus shows the same difference between abstractive and forced extractive generation modes while having the same level of extractiveness as BART and PRIMERA. On the other hand, these results imply that BART doesn't understand the concept of iterative sentence extraction and performs summarization by fusing individual fragments. The robustness of PRIMERA indicates high confidence in generation strategy, which may

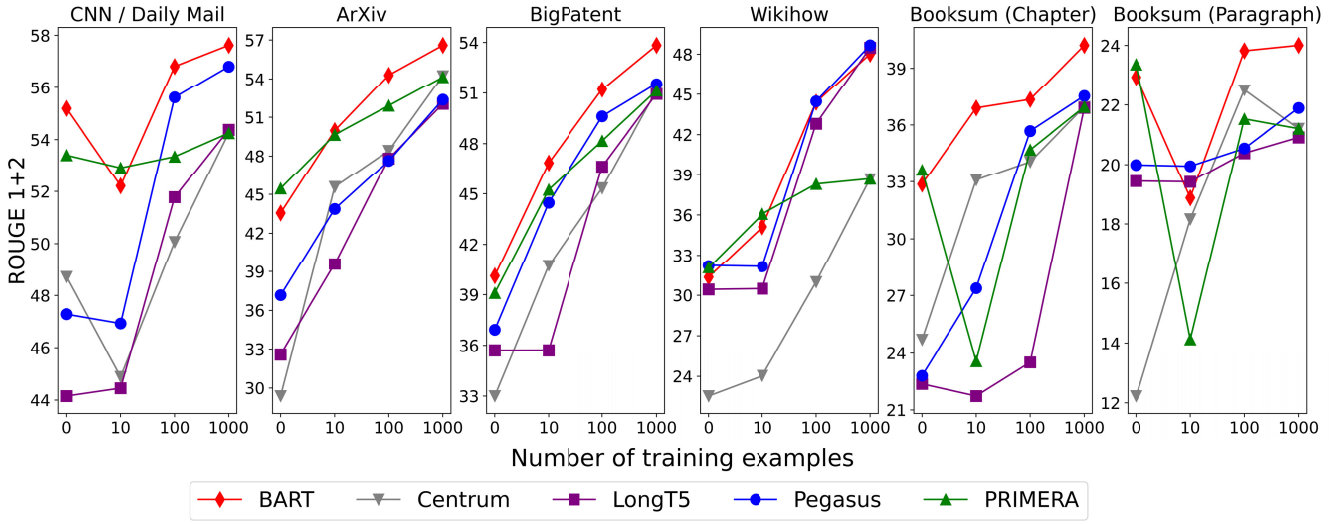

FIGURE 2. ROUGE-1 + ROUGE-2 F1 scores of pre-trained models for all low-resource settings.

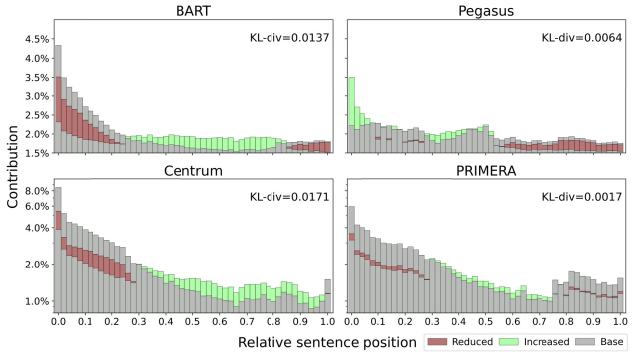

FIGURE 3. Average ALTI sentence-wise contribution distribution dynamic throughout few-shot 1000 tune.

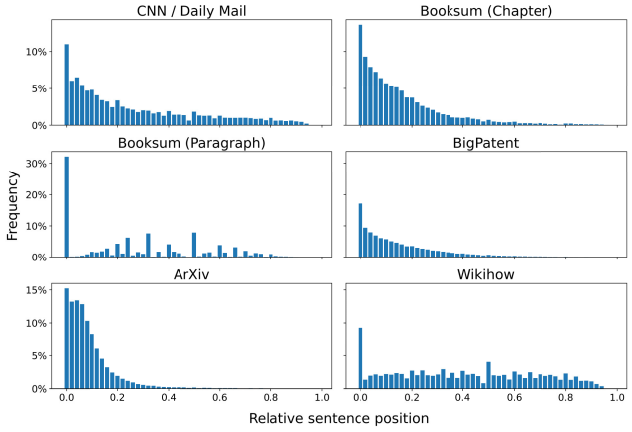

FIGURE 4. Average Extractive Oracle sentence distribution for all datasets.

explain worse adaptation capabilities in unfamiliar domains (e.g. Booksum). The Centrum performance suggests that the model correctly identifies the salient sentences during abstractive summarization but prefers to paraphrase them instead of directly copying.

Table 5 shows Relevant Attention Evaluation results. We don't provide analysis for LongT5 model due to the incompatibility of transient global attention meta-tokens with ALTI attention block decomposition. In contrast to forced extractive setting results, Pegasus shows the lowest attention overlap with the Oracle sentences, while Centrum and BART rank in the first place. However, after tuning the models on 1000 examples Pegasus closes the gap with BART and comes second. PRIMERA on the other hand struggles to improve RELAE which given the Centrum scores suggests a strong pre-training bias.

Figure 3 provides a more detailed examination of ALTI sentence-wise contribution distribution. All pre-trained models but Pegasus are biased towards the beginning of the document. The bias is strongest with Centrum and PRIMERA models that dispatch up to 8% of attention to the initial sentences. At the same time despite the pre-training differences, the attention pattern of those models is very similar, steadily declining with increasing position and spiking at concluding sentences. Considering that only BOS token has access to global attention this pattern might be attributed to limitations of sliding window attention that has a shorter attention span for first and last tokens (half the window size) and thus overestimates their connection with consequent/subsequent tokens. In contrast, Pegasus assumes a uniform sentence saliency distribution even for the concluding sentences.

After tuning on 1000 examples the ALTI distribution changes (colored parts of Figure 3). Pegasus acknowledges the lead bias and pays less attention to the tailing sentences. Centrum and PRIMERA converge to the same contribution distribution and, conversely, shift the attention from beginning to end. BART follows a different trend, discarding the lead bias and replicating Pegasus initial contribution uniformity. The magnitude of ALTI distribution changes (KL

divergence in Figure 3) is the least for PRIMERA model and Centrum is the most unstable. Such a gap between LED models is likely to be related to initial prediction robustness as PRIMERA's attention strategy proved to be efficient in zero-shot setting even for extractive oracle approximation.

ALTI contribution distribution dynamic reflects the sentence position distribution of Extractive Oracle (Figure 4). All datasets exhibit a significant leading sentence bias, which corroborates past analysis [15]. For BigPatent and ArXiv this is attributed to document structure where summarizing information can be found in introduction and conclusion sections (the latter is absent in experiments due to the 1024 token input length limit). The distribution of salient sentences in news articles from CNN / Daily Mail is also due to the popular Inverted Pyramid writing style, which instructs ordering the information according to its relevance. The uniformity of WikiHow dataset is the result of the data collection process that ensures that each paragraph of instruction is represented in a reference summary. Booksum is the dataset with the highest summary abstractiveness, especially at the paragraph level, where the summary is limited to 1-2 sentences (the average is 1.73). The Extractive Oracle cannot deduce a consistent pattern other than extracting the introductory sentences.

Therefore, initial model contribution-data distribution discrepancy plays and important role in zero-shot performance and stability of adaptation process. Comparing the copy rates after tuning on 1000 examples (Table 4) we conclude that the initial attention pattern contributes to the behavior shift of tuned models, with more uniform distributions encouraging more abstractive strategies. At the same time, strong alignment of sentence-wise contribution bias with dataset extractive oracle distribution may lead to early convergence to extractive solutions and hinder further adaptation as PRIMERA has demonstrated.

IV. ALLEVIATING THE PRE-TRAINING BIAS

Throughout our evaluation, BART demonstrated superior task comprehension capabilities as well as great adaptability. Pegasus on the other hand lacks any dataset insights and thus adapts at a much slower rate. At the same time, Pegasus is the only model that avoids performance instabilities in low-resource settings. In this section, we attempt to negate Pegasus shortcomings by following two different approaches: additional pre-training on BART sentence permutation task and encoder attention shift.

A. MERGING PEGASUS WITH BART

BART doesn't have an exclusive summarization pre-training, however, it is heavily implied in the evaluation. We hypothesize that zero-shot summarization capabilities are the side-effect of sentence permutation task and pre-training dataset composition.

BART was trained on the same data as RoBERTa [24] encoder-only Transformer language model which mainly consists of news articles (∼71%, CC-News [25] and OpenWebText[2]). As was mentioned in section III-E, news articles are commonly written in an Inverted Pyramid style, which guarantees that sentences are ordered according to their saliency. In this context sentence permutation becomes equivalent to extractive summarization: the shuffled document acts as a general collection of topics and the original, which we aim to restore, represents their ground-truth saliency ranking, which can be converted to sentence classification labels used in extractive summarization by applying positional threshold (e.g., first 3 sentences or first paragraph). While BART recovers the full text during pre-training and therefore doesn't learn the concept of saliency filtering, there are two factors that implicitly apply the positional threshold during generative inference: user-defined maximum length of generated sequences and the exposure bias [34], that attributes to prediction uncertainty growth with each generated token and thus to early probability space collapse to end-of-sequence (EOS) token.

Although the importance of sentence permutation pre-training for summarization fine-tuning was noted in the original work [20], the authors didn't report any ablation results for low-resource settings. At the same time, they showed that sentence permutation must be combined with masked language modeling tasks to ensure text coherence. Thus, to prove our hypothesis we need to either jointly train a new language model on both pre-training tasks from scratch or apply sentence permutation pre-training to existing masked language modeling generative model. Pegasus is the perfect candidate as pseudo-summarization pre-training (recovering masked sentences) is equivalent to text infilling task (recovering masked text spans) which was used in BART model as the main masked language modeling task.

B. GUIDING TOWARDS THE DATA BIAS

As has been shown in section III-E (Figure 4) the statistical probability of the sentence to be chosen as the relevant is biased towards the specific document positions. However, the analysis of attention distribution revealed that models like Pegasus don't take into account the bias and instead assume uniform saliency distribution.

One way to bias the attention is to use the existing binary attention mask (Binary masking) to filter the least relevant positions. The disadvantage of that approach is the partial loss of secondary information that could reinforce the encoded salient concepts as the attention mechanism wouldn't propagate the signal from masked positions. An improved approach is to modify the attention mechanism to use multiplicative float masks (Float masking) along with the original binary additive mask. This would retain the connectivity of the attention graph and shift the centrality at the same time. Another alternative is the text-to-image result blending approach, encoder mixture [33]. The idea is to interpolate the outputs of the original encoder without masking and the encoder with the bias mask. This way the

[2]https://github.com/jcpeterson/openwebtext

TABLE 6. Results of Pegasus additional sentence permutation pre-training.

Model	# Examples	ROUGE-1	ROUGE-2	ROUGE-L	Decoder copy rate
Pegasus-SP	0	**30.06**	**8.36**	16.84	91.05%
	10	**31.52**	**9.26**	18.41	80.44%
	100	33.57	**10.51**	**20.51**	54.83%
	1000	34.45	11.12	**22.09**	32.87%

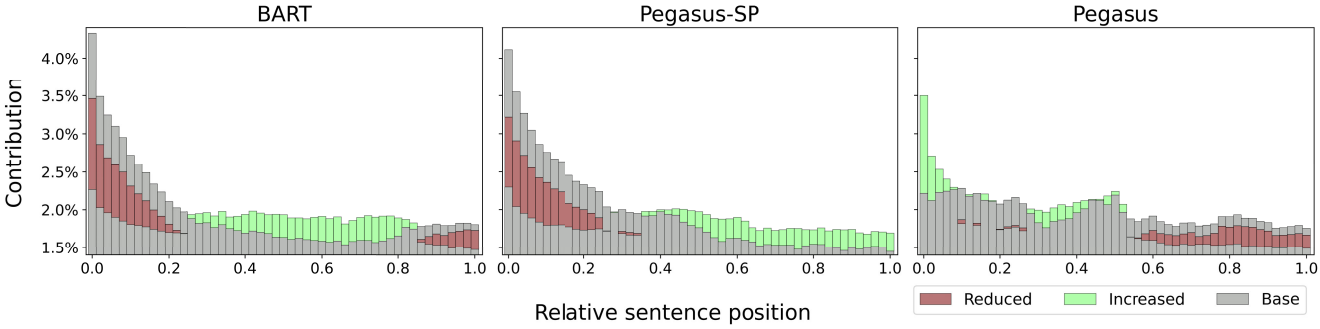

FIGURE 5. Effect of sentence permutation pre-training on attention distribution.

TABLE 7. Comparison of attention shifting methods.

Model	Alteration method	Interpolation	Coeff.	ROUGE 1+2	Rel. improv.
BART	-	-	-	37.67	-
	Binary masking	-	-	37.57	-0.28%
	Float masking	-	-	36.80	-2.31%
	Encoder mixture (original)	lerp	0.45	**38.51**	**2.24%**
		slerp	0.48	38.39	1.91%
	Encoder mixture (identity)	lerp	0.42	**38.51**	**2.22%**
		slerp	0.47	38.30	1.68%
Pegasus	-	-	-	32.74	-
	Binary masking	-	-	32.05	-2.10%
	Float masking	-	-	33.07	1.02%
	Encoder mixture (original)	lerp	0.57	33.49	2.30%
		slerp	0.62	33.59	2.60%
	Encoder mixture (identity)	lerp	0.72	**34.10**	**4.16%**
		slerp	0.48	34.01	3.88%

TABLE 8. Effect of attention shifting methods on Pegasus-SP.

Model	Alteration method	Interpolation	Coeff.	ROUGE 1+2	Rel. improv.
Pegasus-SP	-	-	-	38.42	-
	Binary masking	-	-	38.36	-0.15%
	Float masking	-	-	38.21	-0.54%
	Encoder mixture (original)	lerp	0.43	39.51	2.84%
		slerp	0.53	39.43	2.62%
	Encoder mixture (identity)	lerp	0.47	**39.85**	**3.72%**
		slerp	0.43	39.84	3.69%

bias-masked output would amplify the concepts produced by relevant tokens without disrupting the information flow of the original network.

Unlike other approaches the encoder mixture introduces several hyperparameters: the interpolation coefficient that constraints the decoder to selected tokens, the interpolation method that determines the transformation trajectory, and encoder weights used to produce the biased output. For interpolation we consider linear (lerp) and spherical (slerp). Besides the original encoder (Encoder mixture (original)) we consider an identity version (Encoder mixture (identity)) that encourages the decoder to copy the full input directly.

C. EXPERIMENT SETTING

Our sentence permutation pre-training dataset consists of 367 932 articles published between January 2017 and December 2019 filtered out from CC-News. Since Pegasus is already capable of generating coherent summaries to avoid overfitting we pre-train the model for 3 epochs. We use batch size 32 and Adam optimizer with learning rate = 3e-5 and linear scheduler with warmup ratio = 10%.

To compare the efficiency of the attention correction approaches we test them on the best performing models, BART and Pegasus. To build a multiplicative mask we map oracle distribution weights (section III-E, Figure 4) to token positions of corresponding sentences. The identity encoder is trained on autoencoding task using the original language models with a frozen decoder with the sentence permutation pre-training data and the hyperparameters from section III-D. We bias the auxiliary encoder towards the top-p $\in [10, 70]$ percentile of the most salient positions.

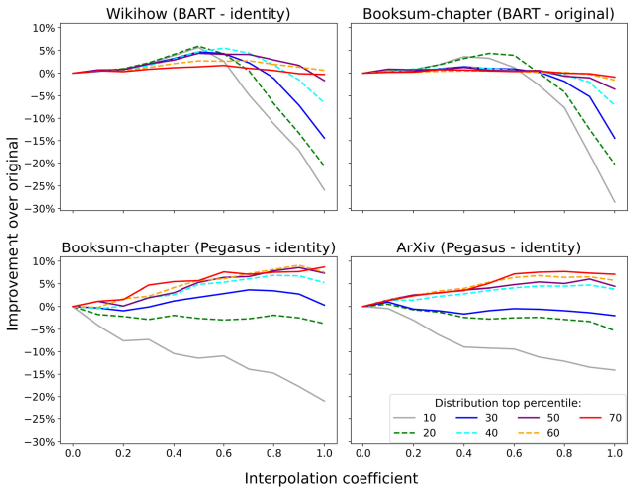

FIGURE 6. Effect of top-p salient sentence filtering in Encoder mixture.

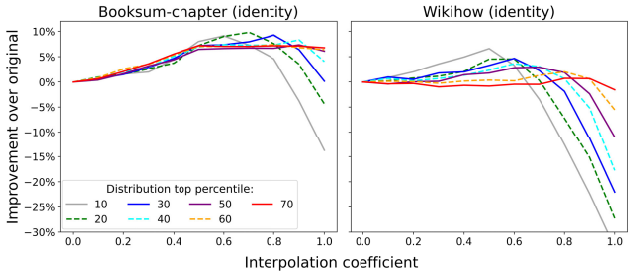

FIGURE 7. Encoder mixture top-p saliency filtering dynamic with Pegasus-SP.

D. MERGING RESULTS

Results of Pegasus additional sentence permutation pre-training (denoted as **Pegasus-SP**) are reported in Table 6. The new model outperforms the alternatives at zero-shot and up to few-shot 100 settings but cedes its leading position to BART at 1000 yet remains superior to the original Pegasus. Interestingly, the source copy rate of generated summaries also increases and follows the same trend as LongT5, which indicates a stronger preference for extractive strategies.

The attention distribution also experiences changes in model tuning trends (Figure 5). Pegasus-SP bears more similarity to BART and exhibits the same lead bias that is suppressed during few-shot 1000 tuning. At the same time, the combined model retains the same distribution tail as Pegasus yet doesn't shift it during tuning. Instead, the model boosts the attention of the second half sentence bringing the distribution to uniformity similar to BART.

Therefore, we confirm that the efficiency of BART in abstractive summarization is a byproduct of the sentence permutation task and specifics of pre-training data which in combination reduce to sentence saliency ranking task. This has two implications. First, length-truncated sentence permutation lies in the solution space of summarization. Second, sentence saliency-based reordering is the basis of the efficient pre-trained abstractive summarization model and centrality-based pre-training (e.g. Principal pseudo-summarization) isn't a sufficient substitute.

E. ATTENTION CORRECTION RESULTS

The average performance of attention shifting approaches is reported in Table 7. Classic attention masking has controversial results. Both methods degrade the summarization quality, however, with multiplicative float mask BART experiences the major drop while Pegasus, on the contrary, gains a slight boost. Encoder mixtures, on the other hand, are more consistent. Linear interpolation proves to be the most effective with at least a 2% relative improvement. Using original or identity encoder for attention-biased input has little effect on BART model. In contrast, the identity encoder for Pegasus has almost the double original encoder's relative improvement achieving a 4% quality increase.

The influence of the auxiliary encoder for BART is less than 50% with an interpolation coefficient of 0.45 indicating that the main non-biased encoder has a good salient content intuition, which is expected considering previously discovered alignment of attention patterns. On the other hand, Pegasus given the lack of attention alignment tends to replace the signal of the main encoder by more than 70%. Considering that this is the case of identity auxiliary encoder that encourages copying text directly this fact suggests that the original encoder tends to introduce additional context information that drives the decoder away from the input text during the summary generation process.

Figure 6 gives a closer look at extreme cases of encoder mixture improvement. The largest quality gains are seen on data with the most discrepancy between attention and oracle sentence distribution. For WikiHow and Booksum-chapter BART improves by about 5% with encoder bias set to top-20% of oracle positional distribution. Pegasus encoder mixture has the best results on Booksum-chapter and ArXiv with over 8% improvement using a much softer top-60% bias mask which is equivalent to suppressing positions corresponding to the attention distribution tail. The interpolation coefficient however exceeds 80% meaning that Pegasus shifts its behavior from central sentence estimation to document lead extraction.

The analysis of Pegasus-SP showed that the improved model exhibits the same attention distribution pattern as BART and thus should also benefit from attention correction. Table 8 reports the results of combined bias correction approaches. Contrary to the patterns, the improved model has the same attention-shifting sensitivity as the original Pegasus, benefiting the most from the identity encoder mixture method. However, the interpolation coefficients are in the same range as BART's and the original encoder signal remains dominant, indicating a much better encoder information propagation in extractive cases.

Sentence permutation pre-training also influences the top-p sampling dynamic (Figure 7). In extreme cases, Pegasus-SP no longer diverges at lower percentile values and follows the same pattern as BART. The model now majorly benefits from attention correction on WikiHow dataset as the attention distribution is less uniform. However, the magnitude of ROUGE 1+2 improvements is much higher, with

∼10% and ∼7% relative improvements on Booksum-chapter (top-20%) and WikiHow (top-10%) respectively.

V. CONCLUSION

In this work, we studied the reasons behind substantial performance differences between summarization-specialized (Pegasus, LongT5, Centrum, PRIMERA) and general-purpose pre-trained generative language models (BART) in low-resource settings. Our probing experiments found that while all models exhibit extractive behavior at zero-shot setting those that are pre-trained on a pseudo-summarization task only lack the prediction confidence to extract longer fragments and thus largely benefit from decision-constrained generation mode. At the same time, the attention distribution of the said models doesn't reflect the structure of the real-world texts, assuming a uniform sentence saliency distribution. The latter happens to be the major factor in underperformance in news, fiction, and scientific articles domains as summaries for those texts are usually biased towards the leading sentences.

We revealed that the superiority of BART general-purpose generative model throughout all low-resource settings is the result of a combination of sentence permutation pre-training task and specificities of the pre-trained data that reduces to saliency-guided sentence reordering. This hypothesis was confirmed by additionally pre-training Pegasus on sentence permutation task with the subset of BART's pre-training data. The evaluation results demonstrated that the new model bears more similarity to BART in both prediction quality and attention distribution and outperforms the original Pegasus in all low-resource settings. To prove the importance of attention distribution we tested three attention shifting methods for zero-shot models. Among them the encoder mixture proved to be the most effective, bringing up to 10% quality increase for models and datasets with the largest discrepancy between attention and sentence saliency distribution.

ACKNOWLEDGMENT

The research was carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University.

REFERENCES

[1] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4190–4197.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015, pp. 1–15.

[3] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020, *arXiv:2004.05150*.

[4] A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian, "A discourse-aware attention model for abstractive summarization of long documents," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2018, pp. 615–621.

[5] D. Dale, E. Voita, L. Barrault, and M. R. Costa-Jussá, "Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity even better," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 36–50.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Jun. 2019, pp. 4171–4186.

[7] J. DeYoung, S. C. Martinez, I. J. Marshall, and B. C. Wallace, "Do multi-document summarization models synthesize?" 2023, *arXiv:2301.13844*.

[8] J. Ferrando, G. I. Gállego, and M. R. Costa-Jussa, "Measuring the mixing of contextual information in the transformer," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 8698–8714.

[9] J. Ferrando, G. I. Gállego, B. Alastruey, C. Escolano, and M. R. Costa-Jussa, "Towards opening the black box of neural machine translation: Source and target interpretations of the transformer," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 8756–8769.

[10] D. G. Ghalandari, C. Hokamp, N. T. Pham, J. Glover, and G. Ifrim, "A large-scale multi-document summarization dataset from the Wikipedia current events portal," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1302–1308.

[11] J. A. Gonzalez, A. Louis, and J. C. K. Cheung, "Source-summary entity aggregation in abstractive summarization," in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 6019–6034.

[12] T. Goodwin, M. Savery, and D. Demner-Fushman, "Flight of the PEGASUS? Comparing transformers on few-shot and zero-shot multi-document abstractive summarization," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 5640–5646.

[13] M. Guo, J. Ainslie, D. Uthus, S. Ontanon, J. Ni, Y. H. Sung, and Y. Yang, "LongT5: Efficient text-to-text transformer for long sequences," in *Proc. Findings Assoc. Comput. Linguistics, NAACL*, 2022, pp. 724–736.

[14] J. Li, L. Liu, H. Li, G. Li, G. Huang, and S. Shi, "Evaluating explanation methods for neural machine translation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 365–375.

[15] T. Jung, D. Kang, L. Mentch, and E. Hovy, "Earlier isn't always better: Sub-aspect analysis on corpus and system biases in summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3324–3335.

[16] M. Koupaee and W. Y. Wang, "WikiHow: A large scale text summarization dataset," 2018, *arXiv:1810.09305*.

[17] W. Kryscinski, N. S. Keskar, B. McCann, C. Xiong, and R. Socher, "Neural text summarization: A critical evaluation," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 540–551.

[18] W. Kryscinski, N. Rajani, D. Agarwal, C. Xiong, and D. Radev, "BOOKSUM: A collection of datasets for long-form narrative summarization," in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2022, pp. 6536–6558.

[19] F. Ladhak, E. Durmus, H. He, C. Cardie, and K. McKeown, "Faithful or extractive? On mitigating the faithfulness-abstractiveness trade-off in abstractive summarization," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 1410–1421.

[20] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.

[21] X. Li, G. Li, L. Liu, M. Meng, and S. Shi, "On the word alignment from neural machine translation," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1293–1303.

[22] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3730–3740.

[23] Y. Liu, P. Liu, D. Radev, and G. Neubig, "BRIO: Bringing order to abstractive summarization," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 2890–2903.

[24] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[25] S. Nagel. (2016). *CC-News*. [Online]. Available: http://web.archive.org/web/20230812191312/ and https://commoncrawl.org/2016/10/news-dataset-available/

[26] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, 2016, pp. 280–290.

[27] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, and R. Avila, "GPT-4 technical report," 2023, *arXiv:2303.08774*.
[28] Y. Otmakhova, K. Verspoor, T. Baldwin, A. J. Yepes, and J. H. Lau, "M3: Multi-level dataset for multi-document summarisation of medical studies," in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2022, pp. 3887–3901.
[29] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2018, pp. 2227–2237.
[30] D. Pruthi, M. Gupta, B. Dhingra, G. Neubig, and Z. C. Lipton, "Learning to deceive with attention-based explanations," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4782–4793.
[31] R. S. Puduppully, P. Jain, N. Chen, and M. Steedman, "Multi-document summarization with centroid-based pretraining," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 128–138.
[32] C. Raffel, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.
[33] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with CLIP latents," 2022, *arXiv:2204.06125*.
[34] M. Ranzato, "Sequence level training with recurrent neural networks," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016, pp. 1–16.
[35] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 379–389.
[36] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1073–1083.
[37] E. Sharma, C. Li, and L. Wang, "BIGPATENT: A large-scale dataset for abstractive and coherent summarization," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2204–2213.
[38] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3319–3328.
[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[40] J. Wang, Y. Liang, F. Meng, B. Zou, Z. Li, J. Qu, and J. Zhou, "Zero-shot cross-lingual summarization via large language models," in *Proc. 4th New Frontiers Summarization Workshop*, 2023, pp. 12–23.
[41] M. Wilber, W. Timkey, and M. Schijndel, "To point or not to point: Understanding how abstractive summarizers paraphrase text," in *Proc. Findings Assoc. Comput. Linguistics*, 2021, pp. 3362–3376.
[42] W. Xiao, I. Beltagy, G. Carenini, and A. Cohan, "PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 5245–5263.
[43] J. Xu and G. Durrett, "Dissecting generation modes for abstractive summarization models via ablation and attribution," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 6925–6940.
[44] J. Xu, S. Desai, and G. Durrett, "Understanding neural abstractive summarization models via uncertainty," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 6275–6281.
[45] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 11328–11339.
[46] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKeown, and T. B. Hashimoto, "Benchmarking large language models for news summarization," 2023, *arXiv:2301.13848*.
[47] Y. Zou, K. Song, X. Tan, Z. Fu, Q. Zhang, D. Li, and T. Gui, "Towards understanding omission in dialogue summarization," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2023, pp. 14268–14286.
[48] Y. Chen, Y. Liu, R. Xu, Z. Yang, C. Zhu, M. Zeng, and Y. Zhang, "UniSumm and SummZoo: Unified model and diverse benchmark for few-shot summarization," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2023, pp. 12833–12855.

DANIIL CHERNYSHEV is currently pursuing the Ph.D. degree in physico-mathematical sciences with Lomonosov Moscow State University, Moscow, Russia.

He is an Intern Researcher with the Research Computing Center, Lomonosov Moscow State University. His research interests include automatic summarization, reliable language modeling, and deep learning.

BORIS DOBROV received the Ph.D. degree in physico-mathematical sciences from Lomonosov State University, Moscow, Russia, in 1988.

Since 1999, he has been the Chief of the Laboratory of Information Resource Analysis, Research Computing Center, Lomonosov Moscow State University. He is the author of more than 120 publications in the field of information retrieval and ontologies. His research interests include information retrieval, automatic summarization, text mining, and deep learning.