

Evaluating the Summarization Comprehension of Pre-Trained Language Models

D. I. Chernyshev^{1*} and B. V. Dobrov^{1**}

(Submitted by E. E. Tyrtysnikov)

¹*Research Computing Center, Moscow State University, Moscow, 119991 Russia*

Received March 22, 2023; revised April 15, 2023; accepted April 30, 2023

Abstract—Recent advances in abstractive summarization demonstrate the importance of pre-training tasks, however, general purpose language models manage to outperform summarization-specialized pre-training approaches. While several works addressed the question of pseudo-summarization pre-training efficiency in abstractive summarization fine-tuning, none has explored the properties of pre-trained models in a low-resource setting. This work attempts to fill this gap. We benchmark 5 state-of-the-art pre-trained language models on 5 single-document abstractive summarization datasets of different domains. To probe the models, we propose 4 novel task comprehension tests that evaluate the main components of summarization models. Our experiments reveal that pseudo-summarization pre-training biases the models towards more extractive behavior and inhibits their ability to properly filter the salient content, leading to worse generalization.

DOI: 10.1134/S1995080223080115

Keywords and phrases: *abstractive summarization, neural networks, machine learning.*

1. INTRODUCTION

Text summarization is a problem of text compression that aims to produce the most informative version of the original text with only salient content remaining. There are two approaches to the problem: extractive and abstractive. While extractive summarization leverages existing text fragments to collect the summary, an abstractive approach complements this process with additional language transformations that enable the generation of the most concise texts. The abstractive approach saw the rise of research development with the emergence of powerful sequence-to-sequence neural networks [1] and rapidly advanced since the introduction of pre-trained Transformer-based language models [2] thanks to drop in fine-tuning computational costs attributed by better generalization of pre-trained models. Most recent improvements were aimed at overcoming length limitations and tweaking the pre-training procedure for summarization task [3]. The pseudo-summarization pre-training approach [4] became the basis of state-of-the-art summarization-specialized language models as it demonstrated a substantial improvement in low-resource summarization setting. However, the latest benchmarks [5, 6] indicate that the pseudo-summarization pre-training may lead to suboptimal performance as models pre-trained on general denoising language modeling tasks like BART [14] show better fine-tuning results.

The question of pseudo-summarization pre-training efficiency has been raised in multiple works [7–9], however, the evaluation was limited to specific domains and text quality metrics. The properties of fine-tuned models were studied in terms of text coherence and factual consistency [5, 10–12] and it was confirmed that the pre-training routine impacts these summarization quality characteristics. On the other hand, none of the studies explored the properties of zero-shot models or the logic behind the successful few-shot adaptation.

*E-mail: chdanorbis@yandex.ru

**E-mail: dobrov_bv@mail.ru

In this work, we investigate the factors of computationally efficient abstractive summarization training. We benchmark five state-of-the-art pre-trained language models on five single-document abstractive summarization datasets of different domains in few-shot tuning settings. To address the lack of summarization-specialized probing metrics we propose four novel task comprehension tests that evaluate the efficiency of individual model components. To study the applicability aspect, we conduct the computational performance evaluation. Our experimental results confirm that denoising pre-training is superior to pseudo-summarization with BART achieving the best performance in all few-shot settings. The examination of model properties reveals that the original Pegasus [4] pseudo-summarization pre-training approach biases the models towards extractive behavior and suppresses the ability to detect salient content, while BART exhibits a more abstractive generation logic and properly exploits the extractive patterns.

2. RELATED WORK

2.1. *Abstractive Summarization*

With the introduction of pre-trained language models with global context knowledge like BERT [2], abstractive summarization saw a breakthrough in text generation quality. Liu et al. [13] were one of the first to propose a successful approach to employ BERT encoder for text summarization tasks with BERTSum model and dual-stage training. While BERTSum surpassed the previous RNN models the full potential of the language model was not utilized due to an undertrained Transformer decoder that wasn't conditioned to apply global contextual knowledge accumulated in the encoder. Lewis et al. [14] solved the issue by employing denoising pre-training in BART language model which allowed it to be fine-tuned directly to various text generation tasks, including abstractive summarization. BART significantly improved previous results and proved to be efficient even in few-shot setting, however, Zhang et al. [4] argued that abstractive summarization required an exclusive pre-training regime. They proposed using pseudo-summaries as a summary proxy for summarization pre-training which was used to train Pegasus abstractive summarization language model. Following this pre-training procedure Guo et al. [3] repurposed T5 [9] language model for long document summarization. Xiao et al. [7] followed the pyramid evaluation method to extend the Pegasus approach to multi-document summarization and used it to pre-train PRIMERA summarization model. Puduppully et al. [8] improved PRIMERA model by swapping the sentence-wise pseudo-summary construction algorithm with a cluster-based document selection procedure.

2.2. *Model Evaluation*

Besides standard summarization text quality metrics (e.g., ROUGE [21]), the researchers questioned the properties of patterns learned by summarization models. The problem of low abstractiveness of summaries generated recurrent models with copy-attention was studied by Zhang et al. [15]. The authors demonstrated that the summaries generated by these models could be obtained with word-level extractive models. The influence on the quality of content selection of components of recurrent abstractive summarization models was examined by Kedzie et al. [16]. The study revealed that biases present in summarization datasets dominate the training signal in standard abstractive summarization setting hindering the model's ability to select relevant content.

Kryscinski et al. [10] were one of the first to investigate the issue of high factual mistake rate in summaries generated by state-of-the-art summarization systems. The authors concluded that up to 30% of generated summaries contain false or distorted facts and listed the common types of errors made by generative models. This line of work was continued by Maynez et al. [11] who extended the problem of factuality to the problem of faithfulness and introduced a novel classification of hallucinations appearing during the generation process. The authors conducted a large-scale study of abstractive summarizers and discovered that more than 70% of generated summaries contain hallucinated content with more than 50% of that content being inconsistent with the source text. In addition, the results showed that models with pre-trained components are less prone to factual mistakes, however current evaluation protocol cannot identify these errors and thus can lead to wrong conclusions. To address the issue of the incompleteness of evaluation protocol Fabbri et al. [5] re-evaluated available evaluation metrics and benchmarked 23 state-of-the-art summarization models. The human evaluation revealed that fully pre-trained encoder-decoder language models such as BART and Pegasus produce summaries

of the highest quality. While none of the metrics could fully explain human judgment, specific variants of standard summarization evaluation metrics showed a high correlation in terms of summary factual consistency, language fluency, and content relevance.

3. EXPERIMENTAL SETUP

The main objective of this work is to study the practicality of existing pre-training routines. To emulate the real-world scarcity of abstractive summarization datasets we evaluate models at zero-shot and few-shot settings on five datasets of different domains. Standard summary generation quality metrics were proposed before adoption of neural networks and thus don't evaluate the task comprehension capabilities of language models. To fill the gap, we propose four novel summarization comprehension metrics to shed some light on the logic behind the efficiency of low-resource summarization.

3.1. Datasets

To evaluate summarization performance in zero-shot and few-shot settings we consider 5 datasets.

CNN/Daily Mail. Originally the CNN/Daily Mail news dataset was designed for question answering task but was later repurposed by Nallapati et al. [1] for abstractive summarization. Each news story is supplied with a set of human-written highlight sentences that serve as a summarization target. The dataset is considered to be an evaluation standard for summarization tasks thanks to the high summary quality and factuality [17]. Number of examples: 312k.

Arxiv. One of the first large-scale long document summarization datasets for scientific domain [18]. 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. Number of examples: 113k.

BigPatent. A collection of United States patent documents across nine different technological areas with over 1 million document-summary pairs [19]. BigPatent was automatically scraped from Google Patents Public Dataset and similarly to Arxiv, utilizes an abstract section as the summarization target. Number of examples: 1341k.

Wikihow. A large-scale dataset of instructions from the online WikiHow.com website [20]. Each instruction article is presented in 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. Number of examples: 200k.

Booksum. A collection of datasets for long-form narrative summarization of texts from the literature domain [6]. 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. Number of examples: 146k and 12k, respectively.

3.2. Models

BART. One of the first Transformer-based encoder-decoder pre-trained language models specifically designed for natural language generation tasks [14]. BART was pre-trained on four text denoising tasks: masked language modeling, omitted language modeling (no [MASK] tokens), sentence order recovery, and token inverse rotation. In previous works, BART demonstrated a good performance in abstractive summarization tasks in both zero-shot and few-shot settings, despite the lack of summarization tasks in the pre-training procedure [6].

Pegasus. One of the first generative language models to employ pseudo-summarization pre-training. Pegasus [4] is pre-trained from scratch on Principle sentence generation task, which involves recovering the sentences with the highest overlap with the rest of the text (pseudo-summaries).

PRIMERA. PRIMERA [7] aimed at eliminating two crucial issues of Pegasus: low input length limit (1024) tokens due to $O(n^2)$ memory requirement of attention mechanism, and frequent irrelevance of sentences chosen by Principle strategy. PRIMERA employs LED encoder-decoder Transformer model with sparse attention initialized with BART weights that enable up to 16000 input tokens. LED attention consists of two separate (parameter-wise) components: local diluted sliding window attention and global token-wise full attention. PRIMERA exploits global attention for encoding different documents in input sequence using <doc-sep> special token. For pseudo-summary construction,

PRIMERA utilizes document clusters and considers a named-entity importance pyramid for more precise sentence saliency evaluation.

LongT5. Similarly to PRIMERA, LongT5 [3] expands T5 (T5.1.1) encoder-decoder pre-trained generative language model for longer documents and employs a sparse attention mechanism. LongT5 uses its own version of global attention, that instead of separating global parameters introduces transient global tokens to local attention by summing the embeddings within each attention window. To repurpose T5 for abstractive summarization LongT5 pre-trains on Pegasus's Principle sentence generation task.

Centrum. Centrum [8] is a variation of PRIMERA model with a different approach to pseudo-summary construction. 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 cluster centroid document as the target summary and the remaining documents as the source.

3.3. Model Evaluation

3.3.1. ROUGE score. In abstractive summarization ROUGE score [21] serves as the standard for summarization quality evaluation and reflects the model's ability to reproduce the reference summary. We measure two variants of ROUGE F1 scores between generated summary and reference: ROUGE- $\{1,2\}$ (word/phrase overlap) and ROUGE-L (longest common subsequence).

3.3.2. Encoder discrimination accuracy. In the generative encoder-decoder language model the encoder aggregates the context information of the input text into embedding that carries the necessary features for the solved task. Since text summarization is essentially a task of text compression which retains core information, the input and output texts should have similar features. By comparing cosine similarities between source text, reference summary (positive), and false reference summary (negative) and measuring the frequency of correct summary ranking we can evaluate the efficiency of the encoder. For the set of input texts $X = \{X_1, X_2, \dots, X_n\}$ and set of respective reference summaries $Y = \{Y_1, Y_2, \dots, Y_n\}$ the encoder discrimination accuracy is defined as

$$EDA(X, Y) = \frac{1}{n} EDA_{ex}(X_i, Y), \quad (1)$$

$$EDA_{ex}(X_i, Y) = \frac{\cos\text{-sim}(Emb(X_i), Emb(Y_i))}{\max_{Y_j \in Y} \cos\text{-sim}(Emb(X_i), Emb(Y_j))}. \quad (2)$$

where $\cos\text{-sim}(A, B)$ is cosine similarity between embeddings A and B , $Emb(A)$ is an embedding transformation of text A with model encoder averaged over tokens. To avoid outliers, we compute this metric with random Y batches of size 10.

3.3.3. Encoder saliency detection. The summarization task requires understanding the saliency of sentences presented in the text. The encoder provides such understanding through the same source text embeddings that are used by the decoder to condition the generation process. This implies that the encoder must be sensitive to changes in oracle sentences: the cosine-similarity between embeddings of the original text and the text with omitted salient sentences must be minimal. To detect salient sentences, we use greedy algorithm to solve the following extractive oracle problem:

$$\text{Oracle}(X_i, Y_i) = \arg \max_{X'_i \subset X_i} \text{ROUGE-2}_{F1}(X'_i, Y_i) \quad (3)$$

Thus, the encoder saliency detection has the following formula

$$ESD(X, Y) = \frac{1}{n} \sum_i 1 - \cos\text{-sim}(Emb(X_i), Emb(X_i \setminus \text{Oracle}(X_i, Y_i))) \quad (4)$$

3.3.4. Approximate oracle test. Abstractive summarization does not prohibit extracting existing source text fragments. Since most existing summarization datasets possess high extractive characteristics, it is expected that the summarization model would assign higher decoding probabilities to salient sentences. For each source subsequence $x_k \in X_i$ we obtain its decoding probability as

$$\text{Dec-Prob}(x_k, \theta) = \sum_{j=1}^{|x_k|} \log(P(w_j | w_{<j}, X_i, \theta)), \quad (5)$$

where w_j is j th token in decoding sequence, θ is parameters of summarization model. Given the decoding probabilities we solve the oracle problem using the greedy algorithm

$$\text{ApproxOracle}(X_i, Y_i) = \arg \max_{X'_i \subset X_i} \text{Dec-Prob}(X'_i, \theta). \quad (6)$$

To measure the efficiency of the model's oracle detection capabilities we measure ROUGE for resulting extractive summaries and compare them with true extractive oracle.

3.3.5. Relevant attention evaluation. Attention mechanism plays a crucial role in Transformer layer transformations. In generative models, the decoder attention determines the effect of the individual input tokens by weighting encoder embeddings during conditional probability distribution reconstruction. In encoder-decoder architectures, each decoder attention has two components: self-attention (generation history) and cross-attention (source text embeddings). Thus, the encoder attention layer affects the output of each decoder layer and to establish a relationship between input tokens and output distribution we should track the attention changes throughout the computation graph.

Abnar et al. [22] proposed attention rollout, a method to track attention-wise contextual information contribution for encoder-based Transformer models by building an “attention graph” where nodes represent tokens and hidden representations, and edges attention weights. In this graph, each node in different layers is connected by multiple paths, and edges (attention weights) are corrected with an identity matrix to account for residual connections. To compute the amount of information propagated between nodes in different layers (information flow), the corrected edge weights are multiplied in a recursive manner which produces a relevance matrix

$$R^l = E^l \cdot E^{l-1} \cdot \dots \cdot E^1, \quad (7)$$

$$E^l = 0.5A^l + 0.5I. \quad (8)$$

where A^l is attention matrix of layer l , E^l is edge weights between layer l and $l - 1$. The main issue with the “attention graph” is that it ignores pre- and post- attention linear transformations that redistribute the weights. The full attention transformation was formulated by Kobayashi et al. [23]:

$$\tilde{x}_i = \sum_{j=1}^J T_i(x_j) + b, \quad (9)$$

$$T_i(x_j) = L(F(x_j) + \mathbb{1}_{i=j}x_i), \quad (10)$$

$$F(x_j) = \sum_{h=1}^H W_O^h A_{ij}^h W_V^h x_j. \quad (11)$$

where L is layer norm transformation, W_O, W_V is linear layer weights, b is combined bias terms, H is number of layer attention heads, J is number of input vectors. To recover true attention weights or input vector contributions Ferrando et al. [24] proposed ALTI method which computes Manhattan distance between layer output vector $\tilde{x}_i$ and transformations $T_i(x_j)$ of each input vector

$$C_{ij} = \frac{\max(0, -d_{ij} + \|\tilde{x}_i\|_1)}{\sum_{k=1}^J \max(0, -d_{ik} + \|\tilde{x}_i\|_1)}, \quad (12)$$

$$d_{ij} = \|\tilde{x}_i - T_i(x_j)\|_1. \quad (13)$$

By replacing attention weights A^l with contributions C^l the ALTI method obtains the true relevance matrix. ALTI+ [25] algorithm extends ALTI method to encoder-decoder architecture, complementing the relevance computation with a cross-attention component.

Original ALTI+ implementation does not consider sparse attention encoder variants. To overcome this limitation, we rebuild attention matrices of sparse architectures back to dense representation. We apply ALTI+ algorithm on summaries generated by greedy sampling to examine the ability of pre-trained abstractive summarization models to select relevant content from source input during the generation process. We sum the relevance matrix produced by the last decoder layer D across output

tokens to obtain the total contribution of each input token. To produce sentence relevance scores these contributions are then averaged over sentence tokens to negate the effect of different sentence lengths

$$\text{Attn-Rel}(\text{sent}_k, X, G) = \frac{1}{|\text{sent}_k|} \sum_{j \in \text{sent}_k} \sum_{0 \leq i \leq |G|} R_{ij}^D, \quad R^D \in \text{ALTI}+(X, G). \quad (14)$$

The sentences with relevance scores higher than the upper quartile Q_3 form the salient sentence set. By comparing the set with the extractive oracle, we measure the relevant attention evaluation score (RELAE)

$$\text{RELAE}_{\text{prec}}(X, Y, G) = \frac{|\text{SalientSet}(X, G) \cap \text{Oracle}(X, Y)|}{|\text{SalientSet}(X, G)|}, \quad (15)$$

$$\text{RELAE}_{\text{rec}}(X, Y, G) = \frac{|\text{SalientSet}(X, G) \cap \text{Oracle}(X, Y)|}{|\text{Oracle}(X, Y)|}, \quad (16)$$

$$\text{SalientSet}(X, G) = \{\text{sent}_k \in X \mid \text{Attn-Rel}(\text{sent}_k, X, G) \geq Q_3(\text{Attn-Rel}(\text{sent}_k, X, G))\}. \quad (17)$$

where X, Y, G are input, reference, and generated texts.

3.4. Few-Shot Tune Parameters

For few-shot tuning we follow Xiao et al. [7] settings: 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 $\max(200, \text{total_examples} \cdot 10)$, 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. 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 Transformers¹⁾ library and initialize from large-variant checkpoints (over 400 mil parameters), except for Centrum which is based on LED-base version (~ 150 mil parameters). For a fair comparison, we limit the input length of all models to 1024 tokens. We sample few-shot training examples from the beginning of the training part of each dataset.

4. EXPERIMENTAL RESULTS

4.1. Few-Shot Evaluation

Table 1 shows the performance comparison among all models. Despite multi-document specialization, PRIMERA outperforms its single-document counterparts in a zero-shot setting, however, the performance gap with BART, which is ranked second, is negligible. Centrum is expected to be the weakest since it has 2.5 times fewer trainable parameters than BART model. On the other hand, LongT5's and Pegasus's low performance indicates the inefficiency of Principle pseudo-summarization strategy, which corroborates previous benchmarks [6, 7].

The situation changes at 10 training examples. PRIMERA scores degrade as the model adapts to a single document setting that tends to have shorter summaries. While Centrum was conditioned on document clusters, the lack of trainable parameters restricts the ability to exploit data patterns and thus enables better adaptation, yet insufficient to reach zero-shot top quality. LongT5 struggles to improve which may be attributed to long-document specialization since we limit the input length. At 100 examples the ranking is similar to zero-shot, however, the performance gap among lower-end models is marginal, while Pegasus and PRIMERA significantly lag behind BART. BART demonstrates great scaling abilities with constant improvements for all few-shot settings.

Data-wise ROUGE 1+2 scores are shown in Fig. 1. As can be seen in most cases pre-trained models exhibit linear performance scaling and performance anomalies are linked to a specific domain. Few-shot tuning on news data from CNN/Daily Mail proves to be challenging as none of the models manages to improve at 10 examples. Centrum, despite its lower comprehension power, outperforms Pegasus on ArXiv and BigPatent datasets, confirming the Principle pseudo-summarization strategy suboptimality

¹⁾<https://huggingface.co/docs/transformers>

Table 1. Average ROUGE scores across six datasets

Example nos.	Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE 1+2
0	BART	29.43	8.24	17.10	37.67
	Centrum	22.57	5.85	14.75	28.42
	LongT5	24.49	6.29	15.73	30.78
	Pegasus	26.07	6.66	16.50	32.74
	PRIMERA	29.65	8.19	17.23	37.84
10	BART	31.18	8.80	18.48	39.99
	Centrum	27.39	7.02	16.72	34.41
	LongT5	25.24	6.67	16.58	31.91
	Pegasus	28.35	7.45	18.19	35.79
	PRIMERA	28.86	8.07	17.59	36.93
100	BART	33.86	10.01	20.10	43.87
	Centrum	30.41	8.16	18.46	38.57
	LongT5	29.57	9.24	20.27	38.82
	Pegasus	32.49	9.62	20.72	42.11
	PRIMERA	32.10	9.02	18.82	41.12

hypothesis. The only dataset where Pegasus and LongT5 are superior is Wikihow which is likely linked to the fact that each paragraph has a direct reference sentence match and thus texts have unbiased extraction patterns. Booksum has the lowest ROUGE scores, especially the paragraph variant. This can be explained by an unfamiliar domain that is poorly represented in pre-training collections, as well high level of summary abtractiveness and salient content uniformity inherent to novels and plays that make up the majority of examples.

4.2. Probing the Models

BART has the highest score in the few-shot 100 setting on 5 out of 6 datasets which contradicts the concept of specialized pre-training objectives. To identify the factors behind this phenomenon we conduct four summarization comprehension tests.

4.2.1. Encoder discrimination test. Table 2 shows the results of encoder embedding discrimination evaluation for test examples from all 6 datasets. The second column values indicate the size of the

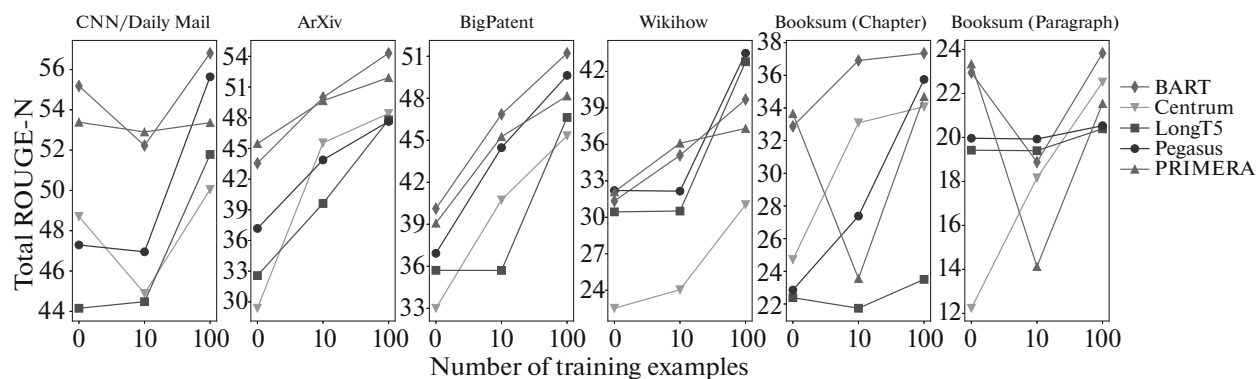
**Fig. 1.** ROUGE 1+2 scores of pre-trained models with 0, 10, and 100 training data.

Table 2. Encoder discrimination evaluation results

Model	EDA	Positive-negative margin size	ESD
BART	0.668	0.247	0.982
Centrum	0.772	0.306	0.980
LongT5	0.630	0.246	0.982
Pegasus	0.657	0.250	0.984
PRIMERA	0.704	0.253	0.986

similarity margin between positive and negative examples. According to Encoder Discrimination Accuracy (EDA), Centrum model has the best understanding of content relevance on the encoder level. Encoder Saliency Detection (ESD) also supports this claim, however, the difference is negligible as models seem to frequently disagree with the importance of extractive oracle sentences. BART, LongT5, and Pegasus have similar scores and while PRIMERA surpasses them in terms of EDA it has the weakest saliency detection capabilities. This result implies that Centrum encoder performance might be explained by both low model parametrization and multi-document pseudo-summarization pre-training setting.

The example-wise distribution of EDA scores is presented in Fig. 2. In the majority of cases, the model is not confident with relevance ranking (low margin) and mixes the variants within the same cosine-similarity ranges. However, judging by the upper quartile Q_3 it is common for Centrum and PRIMERA to correctly rank the candidates. LongT5 and Pegasus on the other hand have the highest variance in EDA score. With BART having better Q_1 EDA scores we assume that this is the result of issues with Principle pseudo-summarization strategy.

4.2.2. Approximate oracle test. The results of the approximate oracle test are reported in Table 3. Contrary to previous results LongT5 and Pegasus have the best salient sentence sampling performance surpassing PRIMERA by a significant margin. BART demonstrates the weakest salient sentence intuition. This can be explained by the lack of pseudo-summarization pre-training that teaches the decoder to select relevant content for sentence restoration. Interestingly, the results of forced extractive generation for LongT5 and Pegasus models are better than the results of zero-shot generation, suggesting that Principle pseudo-summarization encourages extractive behavior. A similar comparison of BART model reveals the high abstractiveness behind the model's generation logic which may be the key to high adaptability. At the same time, PRIMERA has the most balanced strategy as the model performance range remains the same for both summary generation settings.

4.2.3. Relevant attention evaluation. Table 4 shows the results of the relevant attention evaluation. Due to the large difference between the original Transformer and LongT5 layer implementations, we could not rebuild full attention matrices and perform attention rollout within acceptable error boundaries. Therefore, we only report the results for the remaining four models. As can be seen, BART exhibits the

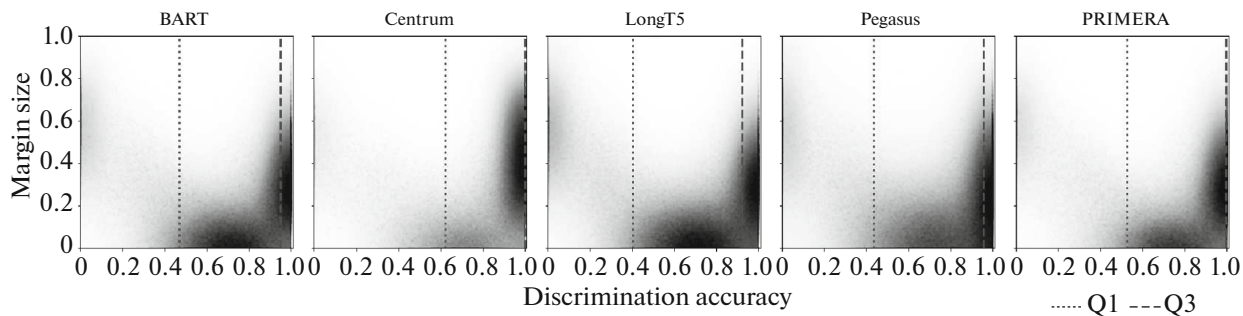


Fig. 2. Encoder discrimination accuracy vs margin size example density plot. Q_1 and Q_3 correspond to upper and lower EDA quartiles.

Table 3. Approximate oracle test results

Oracle scorer	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE 1+2
ROUGE-2 F1	41.60	15.73	23.16	57.33
BART	22.02	4.65	13.00	26.67
Centrum	26.36	5.87	15.20	32.23
LongT5	29.76	7.17	16.62	36.93
Pegasus	29.94	7.21	16.71	37.15
PRIMERA	29.06	6.70	16.10	35.77

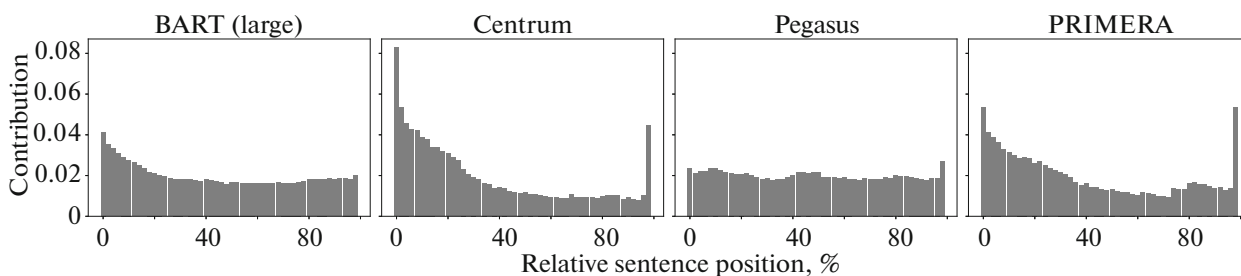
Table 4. Relevant attention evaluation results

Model	RELAE recall	RELAE precision	RELAE F1
BART	36.08	28.77	28.75
Centrum	35.91	28.04	28.15
Pegasus	29.15	23.46	23.46
PRIMERA	32.08	25.07	25.19

highest RELAE metrics meaning that, despite the lowest oracle probability predictions, the model pays enough attention to the salient sentence. This implies that BART scans the text for specific fragments instead of focusing on sentence subset. The second by score model, Centrum, has similar RELAE recall but significantly lower precision, which can be explained by lower parametrization since its larger counterpart, PRIMERA, has a much larger performance gap. Pegasus has the lowest metrics with RELAE F1 difference with PRIMERA of almost 2%.

The examination of RELAE ALTI+ contributions is presented in Fig. 3. Pegasus is the only model with practically uniform contribution distribution, which explains low saliency detection scores. At the same time, this fact calls into question the efficiency of the pre-training task, as Fig. 4 demonstrates that saliency distribution varies from dataset to dataset. The same figure also illustrates the reason behind the superior performance of Pegasus and LongT5 models on Wikihow dataset.

Wikihow displays mostly uniform distribution. For other datasets the saliency distribution is skewed towards the beginning of the text, corresponding to the contribution distributions of BART, PRIMERA, and Centrum models. The paragraph version of Booksum has a multimodal extraction pattern that is, according to dataset authors, attributed to the high level of reference abstractiveness, making the exact oracle sentence matching infeasible. Bimodal contribution distribution of Centrum and PRIMERA models with an additional preference for the last sentences is likely to be the inhibitor of the fine-tuning process that prevents them from achieving BART results.

**Fig. 3.** Sentence-wise ALTI+ contribution distribution.

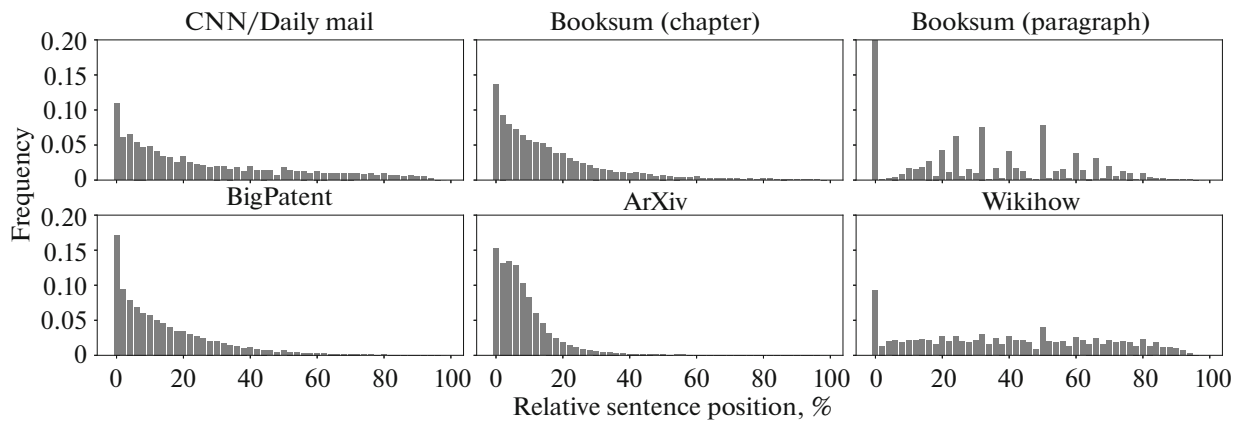


Fig. 4. Extractive oracle relative sentence position distribution.

5. COMPUTATIONAL PERFORMANCE IMPACT

Another important characteristic for practical application is the model tuning costs. We evaluated the tuning performance on Nvidia RTX A4000 which according to Lambda GPU benchmark²⁾ has comparable performance to Tesla V100 in Transformer language model training thanks to hardware-level optimizations of Ampere architecture.

The results of the computational evaluation are presented in Table 5. We measure tuning efficiency as relative ROUGE 1+2 improvement per few-shot 100 tuning session hour and report the time required for convergence to the best score on the validation set. According to the table, Centrum is the most tuning-efficient model with almost a 50% performance increase per hour. The closest competitors, LongT5 and Pegasus, only show 21% and BART has about 17%. The worst tuning efficiency of PRIMERA can be explained by the implementation of a sliding attention mechanism that requires padding the first and the last tokens to a window size of 512 tokens. Considering our input limit of 1024 tokens this means that most of the time the model had to process sequences of maximal length. Another factor that also explains BART results is an inability to identify the inductive bias corresponding to a high zero-shot score, which is implied by performance instability in the few-shot 10 setting on several datasets.

6. CONCLUSIONS

In this paper, we investigated the optimality of pre-training procedures that are designed for abstractive summarization. To study the effects of pre-training routines we proposed four novel summarization comprehension tests: Encoder Discrimination Accuracy, Encoder Saliency Detection, Approximate Oracle Test, and Relevant Attention Evaluation.

We evaluated five state-of-the-art pre-trained generative language models in a few-shot setting. Despite the specialization on multi-document summarization, PRIMERA demonstrated the best

Table 5. Average convergence time and tuning efficiency for 5 models

Model	Few-shot 10 time	Few-shot 100 time	Seconds per epoch	Tuning efficiency
BART	36 m 12 s	57 m 13 s	57 s	0.173
Centrum	34 m 01 s	46 m 58 s	41 s	0.466
LongT5	35 m 18 s	73 m 02 s	52 s	0.215
Pegasus	41 m 14 s	79 m 37 s	56 s	0.217
PRIMERA	32 m 16 s	73 m 08 s	79 s	0.071

²⁾<https://lambdalabs.com/gpu-benchmarks>

results in zero-shot setting, outperforming single-document counterparts. BART comes second and overtakes after a few-shot tuning at 10 and 100 examples while PRIMERA struggles to improve. Centrum proves to be the worst which throughout our experiments is confirmed to be attributed to much lower model parametrization. Our probing experiments reveal the reason behind the underperformance of Pegasus and LongT5 models—the discrepancy of Principle pseudo-summarization strategy to real-world salient sentence distribution. While these models exhibit similar encoder discrimination power to BART and excel at forced extractive generation setting they lack the ability to follow the salient content, possessing uniform token-wise attention distribution. This property renders the models ineffective in real-world texts where sentences in the beginning are more likely to carry the important information. Other models take into account this trait, however, only BART exactly replicates this modality which is the first reason behind high adaptability. Another reason is the lack of extractive oracle intuition revealed in Approximate Oracle test. Considering the best results in Relevant Attention Evaluation, we conclude that BART focuses its attention on the most relevant fragments instead of tracking the full subset of salient sentences which enables a higher level of abstractiveness of text generation logic and thus better generalization. Despite the lower parameter count, Centrum demonstrated similar to BART results in probing tests, suggesting that the model might become the new state-of-the-art in a few-shot setting if scaled to larger sizes. At the same time, the results of computational performance evaluation imply that the current iteration of Centrum is the most cost efficient option and indicate that proper pseudo-summarization pre-training contribute to tuning stability.

FUNDING

The work of Daniil Chernyshev (experiments, survey) was supported by Non-commercial Foundation for Support of Science and Education “INTELLECT”. The work of Boris Dobrov (general concept, interpretation of results) was supported by the Russian Science Foundation (project no. 21-71-30003).

REFERENCES

1. R. Nallapati, B. Zhou, C. d. Santos, C. Gulcehre, and B. Xiang, “Abstractive text summarization using sequence-to-sequence RNNs and beyond,” in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, 2016*, pp. 280–290.
2. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019*, Vol. 1, pp. 4171–4186.
3. 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 *Findings of the Association for Computational Linguistics, Proceedings of the NAACL 2022 (2022)*, pp. 724–736.
4. J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, “PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization,” in *Proceedings of the 37th International Conference on Machine Learning (2020)*, pp. 11328–11339.
5. A. R. Fabbri, W. Kryscinski, B. McCann, C. Xiong, R. Socher, and D. Radev, “SummEval: Re-evaluating summarization evaluation,” *Trans. Assoc. Comput. Linguist.*, 391–409 (2021).
6. W. Kryscinski, N. Rajani, D. Agarwal, C. Xiong, and D. Radev, “BookSum: A collection of datasets for long-form narrative summarization,” *arXiv: 2105.08209* (2021).
7. W. Xiao, I. Beltagy, G. Carenini, and A. Cohan, “PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Vol. 1: Long Papers (2022)*, pp. 5245–5263.
8. R. Puduppully and M. Steedman, “Multi-document, summarization with centroid-based pretraining,” *arXiv: 2208.01006* (2022).
9. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.* **21** (1), 1 (2019).
10. W. Kryscinski, B. McCann, C. Xiong, and R. Socher, “Evaluating the factual consistency of abstractive text summarization,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing EMNLP (2020)*, pp. 9332–9346.
11. J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, “On faithfulness and factuality in abstractive summarization,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020)*, pp. 1906–1919.

12. D. Chernyshev and D. Boris, "ClusterVote: Automatic summarization dataset construction with document clusters," in *Speech and Computer SPECOM 2022, Proceedings of the Conference*, Lect. Notes Comput. Sci. **13721**, 99 (2022).
13. Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (2019), pp. 3730–3740.
14. 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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020), pp. 7871–7880.
15. F. Zhang, J.-g. Yao, and R. Yan, "On the abtractiveness of neural document summarization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (2018), pp. 785–790.
16. C. Kedzie, K. McKeown, and H. Daume III, "Content selection in deep learning models of summarization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (2018), pp. 1818–1828.
17. P. Tejaswin, D. Naik, and P. Liu, "How well do you know your summarization datasets?" in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (2021), pp. 3436–3449.
18. 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 *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 2: *Short Papers* (2018), pp. 615–621.
19. E. Sharma, C. Li, and L. Wang, "BIGPATENT: A large-scale dataset for abstractive and coherent summarization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019), pp. 2204–2213.
20. M. Koupae and W. Y. Wang, "WikiHow: A large, scale text summarization dataset," arXiv: 1810.09305 (2018).
21. C.-Y. Lin, "ROUGE: A package for automatic, evaluation of summaries," in *Text Summarization Branches Out* (Assoc. Comput. Linguist., Barcelona, 2004), pp. 74–81.
22. S. Abnar and W. Zuidema, "Quantifying attention, flow in transformers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020), pp. 4190–4197.
23. G. Kobayashi, T. Kuribayashi, and K. I. Sho Yokoi, "Incorporating residual and normalization layers into analysis of masked language models," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (2021), pp. 4547–4568.
24. J. Ferrando, G. I. Gallego, and M. R. Costa-jussa, "Measuring the mixing of contextual information in the transformer," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (2022), pp. 8698–8714.
25. J. Ferrando, G. I. Gallego, 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 *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (2022), pp. 8756–8769.