PROM: A Phrase-level Copying Mechanism with Pre-training for Abstractive Summarization

Xinbei Ma[1,2,∗] **, Yeyun Gong**[3, #, †]**, Pengcheng He**[#]**, Hai Zhao**[1,2,†] **, Nan Duan**[3]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University
[3]Microsoft Research Asia
sjtumaxb@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn,
{yegong, nanduan}@microsoft.com, Herbert.he@gmail.com

Abstract

Based on the remarkable achievements of pre-trained language models in abstractive summarization, the copying mechanism has proved helpful by improving the factuality, stability, and overall performance. This work proposes **PROM**, a new **P**h**R**ase-level c**O**pying **M**echanism that enhances attention on n-grams, which can be applied to zero-shot summarization with pre-training. PROM adds an indicator layer to explicitly pick up tokens in n-gram that can be copied from the source, and calculates an auxiliary loss for the copying prediction. Empirical studies show that PROM makes significant improvements in fine-tuning on benchmarks. In the zero-shot setting, PROM is utilized in the self-supervised pre-training on raw corpora and provides new general baselines on a wide range of summarization datasets. Further analysis shows that PROM performs more reasonable copying and contributes to faithfulness. Our code is publicly available at https://github.com/xbmxb/PROM.

Keywords: Summarisation, Natural language generation, Semi-, weakly- and unsupervised learning.

1. Introduction

The summarization task requires a model to comprehend an input passage and generate a summary. An ideal summary covers the principal information of the source passage, shows consistency and faithfulness, and is fluent as human language (Zhang et al., 2020a; Kryscinski et al., 2020a). Existing summarization strategies can be categorized into two main branches, abstractive (Rush et al., 2015; Nallapati et al., 2016a; Zhou et al., 2017; Zhang et al., 2020a) and extractive (Nallapati et al., 2017; Wang et al., 2019a; Saggion and Poibeau, 2013). Extractive summarization highly relies on extracting salient sentences from the source. Abstractive methods generate output sequences directly from the vocabulary, thus more flexible and closer to humans, but harder to control. Inspired by Transformer-series models (Vaswani et al., 2017), abstractive methods are unified as conditioned seq2seq problem. Language models that have been pre-trained on large-scale corpora (Lewis et al., 2020; Raffel et al., 2020a; Zhang et al., 2020a; Bi et al., 2020; Qi et al., 2020a) dominate this area of research.

The copying method represents a compromise of extraction and generation, alleviating the problems of inconsistency. The consistency or faithfulness of abstractive summarization remains to be improved. Intrinsic reasons lie in the inherent imperfection of models, such as exposure bias (Liu et al., 2022b), insufficient comprehension of the document (Wu et al., 2021; Dou et al., 2021), while extrinsic reasons may be because the excessive confidence of the language model, leading to unfaithful summaries (Chen et al., 2022). The copying method computes a copying distribution on the source sequence, and then aggregates the copying distribution and the language model distribution. Thus, unfamiliar tokens can be directly copied or ignored (e.g. new entities or out-of-vocabulary words) (See et al., 2017; Li et al., 2021).

Summarization also has to face the data bottleneck. On the one hand, high-quality summaries are usually human-generated (Nallapati et al., 2016a; Narayan et al., 2018; Koupaee and Wang, 2018), but human writing shows diversity. On the other hand, language models require a large amount of data for supervised fine-tuning. Copying methods allow an alternative to picking up tokens from the source sequence, coping with expressions which the model is unfamiliar with. Intuitively, such an alternative agrees with the zero-shot situation or domain transfer.

In this paper, we first propose a novel copying model with **PROM** (**P**h**R**ase-level c**O**pying **M**ethod) to enhance the attention of n-grams. Then we further propose a pre-training for zero-shot summarization. Transformer (Vaswani et al., 2017) is our backbone model, on which all our methods are implemented. Existing studies have indicated that

∗ Work done during an internship at [3]Microsoft Research Asia.

# Equal contribution. †Corresponding author.

This paper was partially supported by Joint Research Project of Yangtze River Delta Science and Technology Innovation Community (No. 2022CSJGG1400).

LREC-COLING 2024, pages 13103–13119
20-25 May, 2024.

(i) The language model contributes more to function words than entity words (Chen et al., 2022); (ii) BART(Lewis et al., 2020) has a tendency to copy sentences from the source (Zhang et al., 2022). Thus, n-gram granularity can deserve much more attention for copying than random tokens. Instead of considering entities only (Chen et al., 2022; Xiao and Carenini, 2022), the enhancement needs to be extended to n-grams as they not only contain expressions of language but are also adaptive or common cross domains. Different from previous variants of copying module (Xu et al., 2020; Xiao and Carenini, 2022; Li et al., 2021), we enhance the overlapped phrases and add an explicit loss. Experiments of supervised fine-tuning have proved the advantages of PROM compared to other copying methods. Then we pre-train our model with PROM on raw corpora, to leverage PROM on zero-shot setting. Please note that the process of data construction leverages no characters of downstream testing tasks (Yang et al., 2020a; Fabbri et al., 2021a; Zhao et al., 2022; Bražinskas et al., 2020). Thus, our approach is supposed to be general across datasets or domains. Our pre-trained model contributes new zero-shot baselines on various widely used benchmarks, and achieves comparable scores against previous methods that are more domain-oriented.

Contributions of our work are three folds:

(i) A copying method PROM that enhances copying attention of n-grams. Significant advantages have been proved with empirical studies on supervised fine-tuning.

(ii) A general self-supervised pre-training method that integrates PROM. Our pre-trained model provides new, widely ranged baseline scores in the zero-shot setting.

(iii) A detailed discussion from aspects of faithfulness, human evaluation, data bias, and comparison with large-scale language models. Our model shows higher similarity to reference summaries, better factuality towards input passages, and scalability for various domains.

2. Related Work

2.1. Copying Mechanism

In seq2seq tasks, the copying mechanism allows the model to directly look at the source sequence during the generation. In addition to selecting a token from the vocabulary as the next one, picking a token from the source to copy is provided as an alternative. Building a bridge between the predicted token and the source sequence, the copying mechanism plays an important part in summarization. COPYNET (Gu et al., 2016) first introduced the copying mechanism into the seq2seq framework. Simulating the rote memorization as humans will do, COPYNET shows effectiveness on short text summarization (Hu et al., 2015) and single-turn dialogue response. Pointer-generator (See et al., 2017) controls the copying mechanism by a calculated generation probability, leading to prominent progress on summarization benchmarks (Nallapati et al., 2016b; Bi et al., 2020).

Recent studies bring up improved variants of copying. Bottom-up attention (Gehrmann et al., 2018) trains an extra content selector as a hard mask of copying distribution, where a threshold is used to filter source tokens. SAGCopy (Xu et al., 2020) deploys an attention graph for modeling relations among tokens, thus also improving the copying distribution. SeqCopyNet (Zhou et al., 2018) predicts an end position for the copying at each time to suggest a span to be extracted. For multi-document summarization, a copying module helps with preserving details and handling rare tokens for respective docs (Brazinskas et al., 2020). For better factual consistency (Kryscinski et al., 2020b), named entities in source sequence can be incorporated into the vocabulary, thus can be directly copied as an atomic unit (Xiao and Carenini, 2022). Coconet (Li et al., 2021) models the correlations of source tokens from semantic and positional perspectives and uses them to weight the copying distribution. The copying distribution is also fused with former values to make the generator aware of previous states.

2.2. Low-Resource Summarization

As high-quality summarization datasets are extremely expensive to acquire from the natural world (Hua and Wang, 2017; Wang et al., 2019b), the ability to adapt to multiple domains is expected on a well-trained model, specifically, in the few-shot or zero-shot setting. Pre-training is the most widely explored method for general improvements in multiple domains (Gururangan et al., 2020). Transformer-based language models that are pre-trained on large-scaled corpora (Lewis et al., 2020; Raffel et al., 2020b) achieve fluent and reasonable summaries for a wide range of datasets. Towards summarization task, specialized training objectives and training strategies even go a step further (Zhang et al., 2020b; Qi et al., 2020b; He et al., 2022).

Recent studies carefully utilize task-oriented adaptation methods for better scores. Adaptsum (Yu et al., 2021) fills the gap between pre-trained BART (Lewis et al., 2020) and data of specific domains. The second pre-training methods provide benchmarks on low-resource domains like dialogue, social media, etc (Dinan et al., 2019; Gliwa et al., 2019; Rashkin et al., 2019; Zhang et al., 2018; Kim et al., 2019). TED (Yang et al., 2020a) is pre-trained on pseudo summaries constructed from news and transferred within news domains only, using the methods of theme modeling and denoising. Subsequently, WikiTransfer (Fabbri et al., 2021b) simulates features of the target tasks from aspects of extractive diversity and compression ratio. This task-oriented method leads to significant improvement, but also requires specialized pseudo data and models for each target task. Prompt learning (Liu et al., 2022a) is also introduced into the few-shot summarization. Following prompt pre-training, the fine-tuning uses only 300 samples and balances the effectiveness-efficiency trade-off. This work combines the proposed novel copying method PROM with zero-shot summarization by a general pre-training, achieving higher scores than mainstream baselines. More recently, large language models (LLMs) have shown impressive zero- and few-shot abilities (Ouyang et al., 2022; Brown et al., 2020; Chowdhery et al., 2022) and scalability. Input an instruction, the LLM can recognize the intent and give human-like responses. Whereas, the larger scale consumes heavier computational resources. Comparisons are presented in our analyses.

3. Methodology

In this section, after defining the summarization task, we introduce PROM and our pre-training with PROM for zero-shot setting. Summarization can be formulated as a seq2seq task, where an output summary $Y = [y_0, y_1, y_2, ..., y_t]$ is expected given a source sequence (article, news, log) $X = [x_0, x_1, x_2, ..., x_s]$. In the following, the reference summary is denoted as Y while the predicted summary is denoted as Y_{prd}. An overview of our model is shown in Figure 1. PROM follows the Transformer-based encoder-decoder framework and utilizes the copying module to encourage precise and reasonable copying.

3.1. PROM

3.1.1. Backbone

Transformer-based seq2seq models stack attention layers for encoding and decoding, and produce subsequent tokens from vocabulary autoregressively. This process can be denoted as

$$\begin{aligned} H_t^{En} &= F_{En}(X), \\ H_t^{De} &= F_{De}(H_t^{En}, [y_0{:}y_{t-1}]), \\ P_t^{vocab} &= lm(H_t^{De}), \\ \mathcal{L}_{summ} &= \sum_t CE(P_t^{vocab}, y_t), \end{aligned} \tag{1}$$

where lm is the feed-forward language model layer. The loss function is the cross entropy CE between the distribution on vocabulary P^{vocab} and label Y.

Article: Hollywood actor John Cusack is the latest supporter to visit WikiLeaks founder Julian Assange in his continued stay at the Ecuadorian Embassy... Assange has avoided being extradited to Sweden by taking shelter in the Ecuadorean Embassy in London since 2012...
Summary: Hollywood actor is latest supporter to visit WikiLeaks founder Assange. Pictured arriving at the Ecuadorian Embassy where Assange is staying. Assange is avoiding extradition to Sweden by taking shelter in embassy.

Table 1: An example for the copying indicator. The labeled bi-grams (sub-word token) are in blue.

3.1.2. Copying with Phrase Enhancement

With the assistance of the copying mechanism, the predicted distribution becomes a combination of logits on both vocabulary and source tokens (Gu et al., 2016; See et al., 2017; Li et al., 2021). Intuitively, they respectively stand for generating from vocabulary and copying from the source.

$$\begin{aligned} &P_t(w) = p_t^{gen} P_t^{vocab}(w) + p_t^{copy} P_t^{copy}(w), \\ &p^{copy} = 1 - p^{gen}, \end{aligned} \tag{2}$$

where w is for some certain sub-word (token) and FC is short for fully connected layer. p_t^{gen} and P_t^{copy} depend on interactions with the source sequence, mostly by cross attention mechanism in existing work.

However, it is shown that PrLMs may have the over-copying problem with summarization. In consideration of this issue, the copying probability of each source token can be explicitly modeled to make the copying more reasonable. First, we make pseudo-labels for copying in the phrase granularity. Tokens in overlapped n-grams between the source articles and the reference summaries are tagged to be copied. An n-token length window goes through the source sequence X, extracting each n-gram sequence from X. If the same n-gram exists in the target sequence Y, tokens in that window are labeled. C denotes the label sequence and can be formulated as

$$\begin{aligned} &C = [c_0, c_1, c_2, \ldots, c_i, \ldots, c_s], \\ &c_i = \begin{cases} 1, & [c_{i-j}, c_{i-j+n-1}] \quad in \quad Y \\ 0, & otherwise \end{cases}, \\ &n \geq 0, 0 \leq j \leq n. \end{aligned} \tag{3}$$

We present an example of bi-gram tagging in Table 1. Labeling n-grams makes a larger scope than

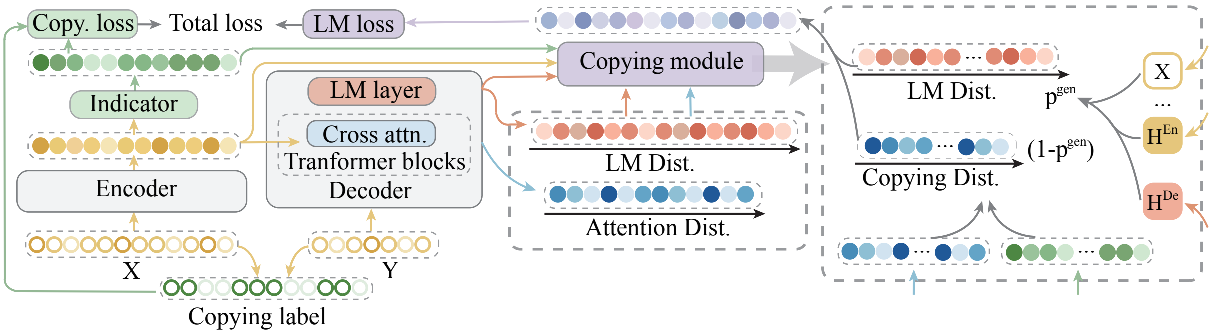

Figure 1: Overview of the proposed PROM. The left part shows the architecture of our model consisting of the Encoder, Decoder, and Copying module, while the right part shows a closer look at the Copying module.

named entities (e.g. *at the Ecuadorian Embassy*), and leaves other tokens like function words to the language model (e.g. the first *is*, the third word in the summary). (Appendix A shows details of this example with analysis.) This method makes intuitive sense as (i) Sequential overlaps are selected and the odd tokens are filtered; (ii) Overlapped n-grams are often more meaningful than uni-grams. Thus, phrases are enhanced in our copying module, like named entities or grammar patterns.

Then we add an indicator layer on the top of the encoder. It is a linear module that takes the encoder hidden states as input and predicts the probability of copying H_C. Then a cross-entropy loss can be computed for the probability H_C and the copying label C.

$$\begin{aligned} H_C &= sigmoid(FC(H_t^{En})), \\ \mathcal{L}_{copy} &= CE(H_C, C). \end{aligned} \tag{4}$$

In our copying module, the copying prediction H_C is integrated to facilitate the copying distribution. Source tokens that have high copying probabilities should be more likely to be selected.

$$\begin{aligned} a_C &= sigmoid(FC(H_C, a)), \\ \tilde{P}_t^{copy}(w) &= \sum_w a_C, \\ p^{gen} &= sigmoid(FC(a \cdot H^{En}, H^{De}, X)), \\ p^{copy} &= 1 - p^{gen}, \\ \tilde{P}_t(w) &= p_t^{gen} P_t^{vocab}(w) + p_t^{copy} \tilde{P}_t^{copy}(w), \end{aligned} \tag{5}$$

where a is cross attention scores of input tokens, and a_C is the scores integrated with H_C. In terms of the training objective, we use the mainstream cross-entropy loss $\mathcal{L}_{summ}$ and also add $\mathcal{L}_{copy}$.

$$\begin{aligned} \mathcal{L}_{summ} &= \sum_t CE(\tilde{P}_t, y_t), \\ \mathcal{L} &= \mathcal{L}_{summ} + \lambda \mathcal{L}_{copy}. \end{aligned} \tag{6}$$

Two training strategies are tried on our model: (i) multi-task method: use the total loss $\mathcal{L}$ as the training objective. (ii) two-stage method: firstly train the copying indicator for several steps, $\mathcal{L}_{copy}$ as the training objective, and then add up $\mathcal{L}_{summ}$ for more steps, $\mathcal{L}$ as the training objective.

4. Pre-training for Few-shot Setting

The copying method builds a bridge for input tokens over the deep stacks of transformer layers, thus helping with OOD (Gu et al., 2016; Li et al., 2021) and improving the stiffness (Chen et al., 2020). Naturally, humans can deal with unfamiliar words by copying without truly understanding them if they infer that the words are important. Motivated by this, we apply our PROM to zero-shot setting.

We propose to pre-train with PROM on the self-supervised objective to leverage our copying module for performance improvement and generalization on the zero-shot setting. Our self-supervised training dataset is constructed from corpora. Formally, let D denote a natural passage in the corpora, which consists of several sentences $D = \{d_0, d_1, d_2, \ldots d_{\hat{s}}\}$. D is processed into pseudo document-summary pairs $(\hat{X}, \hat{Y})$ in the following two ways.

(i) D_{nat}: Given a passage D, we calculate important score $Score(i)$ for each sentence d_i. The $m\%$ top-scoring sentences are selected and deleted from D. Then a pseudo document-summary pair $(\hat{X}, \hat{Y})$ is generated, where $\hat{X}$ is the selected sentences and $\hat{Y}$ is the remaining passage. Both $\hat{X}$ and $\hat{Y}$ keep the original order. This method follows gap sentence generation (GSG) (Zhang et al., 2020a) but uses Extractive Fragments Density (EFD) (Grusky et al., 2018) as the importance score,

$$\begin{aligned} \mathrm{EFD}(x, y) &= \sum_{f \in \mathcal{F}(x,y)} |f|^2 / |x|, \\ Score(i) &= \mathrm{EFD}(d_i, D \setminus \{d_i\}), \end{aligned} \tag{7}$$

where $\mathcal{F}(x, y)$ is overlapped fragments of sequences x, y.

Data	Genre	Size train/test	Len.(words) Doc./Sum.	Ratio
Summerization Datasets				
CNN/DM	news	287k/11k	682/54	14.03
NYT	news	146k/17k	990/79	13.21
BillSum	bill	19k/2k	1219/174	6.25
WikiHow	instruction	168k/6k	580/62	10.96
arXiv	science	203k/6k	4938/220	35.58
XSum	news	204k/11k	361/21	18.25

Table 2: Statistics and characters of datasets.

(ii) D_{chunk}: However, natural articles vary widely in length, and long articles may be truncated and wasted to fit the model width. Thus, we augment our data by setting maximum and minimum limitations on the document sentence number. Passages are chunked by the maximum, and those shorter than the minimum are discarded. Then the same selection is performed on the chunks as (i).

To be consistent with our copying method, we loosely control the extractiveness level of the pre-training data. The data is filtered by a minimum EFD min_{EFD}. Finally, the pre-training dataset is the filtered union of D_{nat} and D_{chunk}. The training objective is the total loss $\mathcal{L}$ in equation 6.

Our model is trained on the pre-training dataset and evaluated on a wide range of downstream tasks in a zero-shot way. Please note that downstream tasks have different characters in the document and summary length, compression ratio, the proportion of novel tokens, etc. But no downstream task-specialized processing is performed on the pre-training data (Yang et al., 2020a; Fabbri et al., 2021a). Thus no information for target data is available, which is against our goal of the generalized zero-shot method.

Lead bias is a common feature of summarization datasets that is often utilized (Fabbri et al., 2021a; Yang et al., 2020b). It is reasonable that the front part contains more primary information of a document. For comparison and analysis, we also create a lead-biased pseudo dataset by extracting the first $\hat{m}$ sentences as summaries and leaving the rest as documents.

5. Experiments

5.1. Dataset

5.1.1. Summarization data

For summarization tasks, our empirical studies rely on a series of datasets that are diverse in genre, size, length, and also extractiveness or abstractiveness level. The statistics and characters are listed in Table 2 and Figure 2.

CNN/DailyMail (Nallapati et al., 2016a) consists of 93k articles from CNN News and 220k articles from Daily Mail News. We use the non-anonymized version (See et al., 2017) by default.

New York Times is derived from published news from New York Times, whose annotation is conducted by experts or hand-verified. We obtain over 174k examples from the corpus for experiments.

BillSum (Kornilova and Eidelman, 2019) contains 22,218 US Congressional bills. The reference summaries are human-written from the Congressional Research Service.

WikiHow (Koupaee and Wang, 2018) releases 230k summarization examples constructed from WikiHow.com, an online knowledge base. The articles are human-written instructions and summaries are combined subtitles.

arXiv (Cohan et al., 2018) is a collection of 113k scientific papers on arXiv.org. Summaries are annotated as the abstracts of the papers while the remaining are the documents to be summarized. It is a representative long-document summarization dataset.

XSum (Narayan et al., 2018) is derived from BBC articles annotated with human-written single-sentence summaries. It consists of 227k samples that are diverse in topics.

Extractiveness & Abstractiveness Level. For the copying method, the level of extractiveness & abstractiveness is an important feature of datasets. To characterize this, three metrics are illustrated in Figure 2. **(i) Extractive Fragments Density** (Grusky et al., 2018) (Eq. 7) describes the density of shared fragments, thus reflects how well the summary can be regarded as a series of extractions. **(ii) Copy Length** (Grusky et al., 2018; Chen et al., 2020) computes the average length of shared fragments, indicating the tendency of continuous copying. **(iii) n-gram Novelty** (See et al., 2017; Sharma et al., 2019) computes the proportion of n-grams that exists in the summary but not in the article. It can be observed that XSum and WikiHow are significantly more abstractive than others. BillSum encourages extraction the most, with the largest EFD and Copy Length. NYT is the most conservative for novel phrases. Overall, the tendency ranking from abstractive generation to extraction is *XSum* $>$ *WikiHow* $>$ *arXiv* $>$ *CNN/DM* $>$ *NYT* $>$ *BillSum*.

5.1.2. Pre-training Corpora

For pre-training, we use the large-scale corpora of ProphetNet (Qi et al., 2020a). It consists of 160Gb English articles collected from news, stories, and web text, and there is no overlap between the corpora and the above summarization datasets. In our implementation, we first conduct experiments on a 29G subset due to the limited resources. Articles of different genres (news, stories, web text) keep

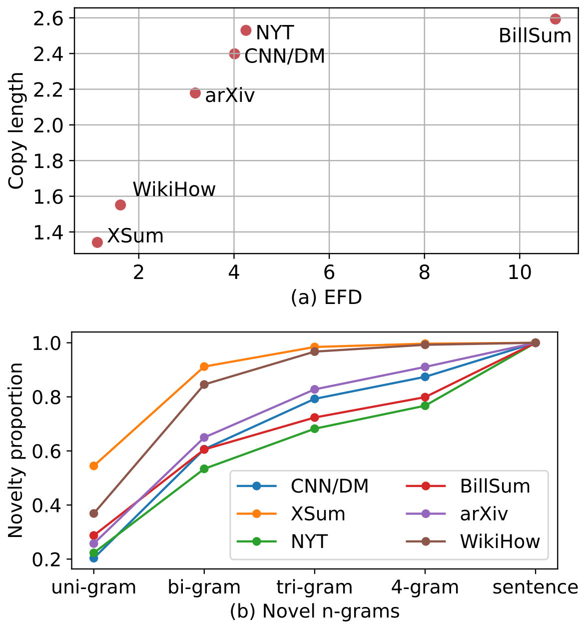

Figure 2: (a) Extractive Fragments Density & Copy Length. (b) n-gram Novelty.

almost the same proportions in the subset as the full-size corpora.

5.2. Experimental Setup

Our empirical study consists of two steps. We first prove the effectiveness of PROM by fine-tuning on summarization datasets. Then we apply PROM to pre-training and evaluate the zero-shot performance, which verifies our motivation to enhance zero-shot summarization with copying.

5.2.1. Fine-tuning Setting

Our model is first fine-tuned to evaluate the effectiveness of PROM. Among the six benchmarks, we choose the most commonly used CNN/DM to compare with previous work (Table 3). We also choose datasets with different characters WikiHow (more abstractive), and arXiv (long documents) to prove the scalability (Table 4).

The previous studies to compare contain commonly used methods and summarization systems related to copying, for example, Point-Generator (See et al., 2017), SAGCopy (Xu et al., 2020), and Bottom-Up (Gehrmann et al., 2018). Notably, CoCoNet (Li et al., 2021) is the previous SOTA of copying methods. These methods all provide results on CNN/DM. For WikiHow and arXiv, the baselines are Lead (See et al., 2017), Point-Generator (See et al., 2017), PEGASUS (Zhang et al., 2020a), and plain BART.

We implement the following settings. In Table 3, three results of BART$_{large}$ are presented. "*Reported*" denotes scores reported in Lewis et al. (2020). "*Fb*" denotes our test results of the released *facebook/bart-large-cnn*. "*Our*" denotes test results of our fine-tuning on *facebook/bart-large*. "*BART$_{large}$+Pointer-Gen.*" denotes integration of BART and copying, i.e., λ in Eq. 6 is 0. For PROM implementations, "*two-stage*" is the second training strategy in Section 3.1.2. "*pre-train*" means fine-tuning after our pre-training.

Our implementation is based on *Transformers* (Wolf et al., 2020). The model is initialized with the official weights of pre-trained BART$_{large}$. The

Model	**R$_1$**	**R$_2$**	**R$_L$**
Previous work			
Lead (See et al., 2017)	40.34	17.70	36.57
Point-Generator (See et al., 2017)	39.53	17.28	36.38
DRM (Paulus et al., 2018)	39.87	15.82	36.90
Bottom-Up (Gehrmann et al., 2018)	41.22	18.68	38.34
S2S-ELMo (Edunov et al., 2019)	41.56	18.94	38.47
DCA (Celikyilmaz et al., 2018)	41.69	19.47	37.92
BERTSUMEXTABS (Liu and Lapata, 2019)	42.13	19.60	39.18
MASS (Song et al., 2019)	42.12	19.50	39.01
SAGCopy (Xu et al., 2020)	42.53	19.92	39.44
UniLM (Dong et al., 2019)	43.33	20.21	40.51
ProphetNet-16G (Qi et al., 2020a)	43.68	20.64	40.72
BART$_{large}$ (Reported)	44.16	21.28	40.90
T5 (Raffel et al., 2020a)	43.52	21.55	40.69
PEGASUS (Zhang et al., 2020a)	44.17	21.47	41.11
ProphetNet (Qi et al., 2020a)	44.20	21.17	41.30
PALM (Bi et al., 2020)	44.30	21.12	41.41
BART+SAGCopy (Li et al., 2021)	44.31	21.35	41.00
CoCoNet (Li et al., 2021)	44.39	21.41	41.05
CoCoPretrain (Li et al., 2021)	44.50	21.55	41.24
Our implementations			
BART$_{large}$ (Our)	43.79	21.20	40.70
BART$_{large}$ (Fb)	44.11	21.08	40.91
BART$_{large}$+Pointer-Gen.	44.11	21.27	40.98
PROM	44.47	21.59	41.32
PROM$_{two_stage}$	44.35	21.61	41.19
PROM$_{pre-train}$	**44.59**	**21.66**	**41.46**

Table 3: ROUGE F_1 scores on CNN/DM.

Model	**WikiHow**			**arXiv**		
	R$_1$	**R$_2$**	**R$_L$**	**R$_1$**	**R$_2$**	**R$_L$**
Lead	24.97	5.83	23.24	28.05	6.63	17.72
Pointer-Gen.	28.53	9.23	26.54	32.06	9.04	25.16
PEGASUS	43.06	19.71	–	44.70	17.27	–
BART$_{large}$	45.22	20.13	43.73	45.18	16.87	39.42
PROM	45.57	20.53	44.09	45.24	16.95	39.38

Table 4: ROUGE F_1 scores on datasets WikiHow and arXiv to show the scalability of PROM.

Model	R_1	R_2	R_L	R_1	R_2	R_L
	CNN/DM			XSum		
Lead	40.34	17.70	36.57	16.30	1.60	12.00
TED	38.38	16.49	35.08	–	–	–
WikiTransfer	40.14	17.71	36.66	31.85	10.44	23.75
w/o bin	39.11	16.98	35.66	22.78	5.66	17.16
w/ GSG	37.62	15.15	34.21	29.95	9.37	21.78
PEGASUS	32.90	13.28	29.38	19.27	3.00	12.72
$\text{PROM}_{\text{subset}}$	37.34	15.26	33.53	22.92	6.30	17.53
$\text{PROM}_{\text{full}}$	37.87	15.91	34.16	22.96	6.05	17.78
	NYT			WikiHow		
Lead	35.50	17.20	32.00	24.97	5.83	23.24
TED	35.03	16.57	31.96	–	–	–
PEGASUS	–	–	–	22.59	6.10	14.44
$\text{PROM}_{\text{subset}}$	36.37	16.45	28.88	25.40	6.04	23.55
$\text{PROM}_{\text{full}}$	36.95	17.21	29.42	25.90	6.37	24.08
	BillSum			arXiv		
Lead	21.09	7.66	18.18	26.46	6.28	22.75
PEGASUS	41.02	17.44	25.24	28.05	6.63	17.72
$\text{PROM}_{\text{subset}}$	39.77	13.88	33.05	34.22	9.29	29.72
$\text{PROM}_{\text{full}}$	40.05	14.66	33.34	34.90	9.71	30.43

Table 5: Zero-shot ROUGE F_1 results of pre-training with PROM. Our results on both subset (Section 5.1.2) and full corpora are reported.

hidden size is 1024 by default. Learning rate is set to {1e-5, 3e-5, 5e-5}, while batch size is set to {36, 48, 64}. The beam search during decoding is in size of 4. For simplicity, n is set to 2, and λ is set to 1. The epoch number is set to {4, 6, 8}, and the best checkpoint is selected by the validation set. More details are shown in the Appendix. Appendix B shows more experimental details.

5.2.2. Zero-shot Setting

In experiments of pre-training for zero-shot setting, our model with PROM is pre-trained on the self-supervised data based on the corpora. Then, the model is tested on all of the six datasets.

The baselines for zero-shot results are Lead (See et al., 2017), TED (Yang et al., 2020a), PEGASUS (Zhang et al., 2020a), and WikiTransfer (Fabbri et al., 2021a). Three settings of WikiTransfer are presented, the best setting, without extractiveness filtering, (w/o bin), and using GSG for summary sentence choice (w/ GSG). Please note that WikiTransfer leverages data features and others are strict zero-shot settings.

We load BART_{large} checkpoints for continuous pre-training on our constructed data. Learning rate is set to {1e-6, 1e-5}, while batch size is {80, 128}. The total epoch number is {2, 4, 6}. Other hyperparameters are the same as those in fine-tuning. Then the model is tested on downstream datasets without fine-tuning. Results are shown in Table 5.

6. Analysis

6.1. Main Results

ROUGE F_1 scores are reported as main results in Table 3, Table 4, and Table 5.

In Table 3, our model surpasses all previous copying methods. PROM leads to significant improvements on each ROUGE score, especially on ROUGE-2. Two-stage training leads to a higher ROUGE-2 score but lower ROUGE-1 and ROUGE-L. Fine-tuned after our pre-training, $\text{PROM}_{pre-train}$ achieves even better performance compared to all related baselines. Table 4 shows the scalability on WikiHow, a more abstractive task, and arXiv, a long-document task. Gains on CNN/DM and WikiHow are larger than on longer sequences.

In terms of the zero-shot results in Table 5, PROM surpasses PEGASUS on most of the datasets except BillSum. This may be because BillSum agrees with over-copying. Compared to methods that utilize domain features, PROM gets lower but comparable scores. Thus, PROM provides a new summarization baseline under the strict zero-shot setting.

To intuitively prove the effectiveness of PROM, we present an illustration of copied contents. We gather n-grams that exist in both the input and the reference (i.e. n-grams ought to be copied), and those that appear in both the input and the predicted summaries (i.e. actually copied), and then compute the F_1 scores. As shown in Figure 3, PROM has advantages on each granularity but is better on bi-gram and longer. The pre-training makes continuous progress. This indicates that PROM corrects the copied contents compared to baselines.

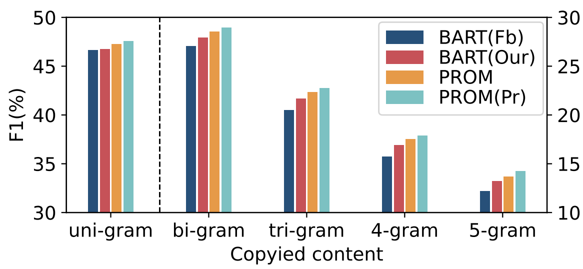

Figure 3: F_1 scores of copied n-grams on CNN/DM. "*PROM(Pr)*" denotes results of $\text{PROM}_{pre-train}$.

6.2. Faithfulness

Recent work shows that PrLM-based abstractive summarization systems still suffer from unfaithfulness (Chen et al., 2022; Wan and Bansal, 2022). With the enhancement of copying, we discuss the faithfulness of our model in the fine-tuning setting.

Part 1: Factuality			
Model	**R-2**	**BERTS.**	**FactCC**
BART_{large} †	21.53	88.36	51.11
BART_{large}(Fb)	21.08	88.31	49.90
BART_{large}(Our)	21.20	88.41	53.70
PEGASUS †	21.47	88.27	50.98
PROM	21.59	88.46	57.39
Part 2: Entity Coverage			
Model	**p_t**	**r**	**F_1**
BART_{large}(Fb)	65.39	74.32	69.57
BART_{large}(Our)	67.29	71.89	69.52
PROM	66.90	74.37	70.44
Part 3: Human Evaluation			
PROM	**Win(%)**	**Tie(%)**	**Lose(%)**
Faithfulness	23.33	56.00	20.67
Informativenss	32.67	34.67	32.67
Readability	16.00	73.33	10.67

Table 6: Advanced metrics. The results are from our models with PROM that are fine-tuned on CNN/DM. The three parts are described in Section 6.2 and 6.3. A dagger means that results are from previous work Chen et al. (2022).

6.2.1. Factuality metrics

Following previous work (Chen et al., 2022; Wan and Bansal, 2022; Pagnoni et al., 2021), we compute two factuality metrics that are highly correlated with human judgment. (i) BERTScore (Zhang et al., 2020c) computes the similarity of contextual embeddings from BERT (Devlin et al., 2019). (ii) FactCC (Kryscinski et al., 2020c) is calculated by a weakly supervised model, which is trained to detect the consistencies and conflicts. The first part of Table 6 presents ROUGE-2 and factuality scores. The higher numbers of PROM show that besides more overlapped bi-grams, the copying method tends to produce more faithful and stable summaries.

6.2.2. Entity Coverage

The consistency of named entities also embodies the faithfulness of a summary (Nan et al., 2021; Xiao and Carenini, 2022; Chen et al., 2022). Hence, we consider entity coverage and present precision, recall, and F_1 between the predictions Y_{prd} and the references Y.

$$\mathrm{p} = |NE\,(Y) \cap NE\,(Y_{prd})|\,/\,|NE\,(Y_{prd})|$$
$$\mathrm{r} = |NE\,(Y) \cap NE\,(Y_{prd})|\,/\,|NE\,(Y)|$$
$$\mathrm{F}_1 = 2 \cdot \mathrm{p} \cdot \mathrm{r}\,/(\mathrm{p} + \mathrm{r})$$

The results are in the second part of Table 6. Our model shows advantages on recall, little difference on precision, but still improves F_1 scores. It suggests that our model generates more related entities, and contributes to faithfulness.

6.3. Human Evaluation

As a complement to the above automatic metrics, we conduct human judgment to evaluate faithfulness, informativeness, and readability. 50 cases are randomly sampled and shuffled, then evaluated by 3 annotators (Wan and Bansal, 2022; Cao and Wang, 2021). Detailed implementation is shown in Appendix C.

The third part of Table 6 shows how PROM performs compared to the BART baseline. We can see that our model significantly wins BART in faithfulness (by 2.66%) and readability (by 5.33%) and ties in informativeness.

In summary, our method not only improves the predicted summaries in terms of the overlap with gold summaries, but also enhances the faithfulness compared to the source document, and better aligns with human preference.

Model	**R_1**	**R_2**	**R_L**	**R_1**	**R_2**	**R_L**
	CNN/DM			**XSum**		
ChatGPT	30.81	11.74	28.54	25.48	8.61	21.54
Llama2-7B	23.89	8.30	21.83	17.30	4.54	14.41
-13B	25.18	9.03	22.94	14.85	2.53	11.85
Llama-30B	25.82	9.38	23.40	15.16	2.68	12.11
$\text{PROM}_{\text{full}}$	37.87	15.91	34.16	22.96	6.05	17.78
	NYT			**WikiHow**		
ChatGPT	32.31	12.70	28.03	21.10	4.30	19.86
Llama2-7B	25.03	8.80	20.90	20.49	5.15	19.53
-13B	28.24	10.95	23.55	21.61	4.95	20.36
Llama-30B	28.44	10.65	23.66	22.20	5.86	20.93
$\text{PROM}_{\text{full}}$	36.95	17.21	29.42	25.90	6.37	24.08
	BillSum			**arXiv**		
ChatGPT	36.57	19.09	33.66	30.95	10.74	27.75
Llama2-7B	31.12	15.21	28.13	23.92	7.65	21.36
-13B	32.14	15.82	29.07	27.80	9.09	24.61
Llama-30B	30.90	14.62	28.00	28.34	8.44	25.10
$\text{PROM}_{\text{full}}$	40.05	14.66	33.34	34.90	9.71	30.43

Table 7: Zero-shot ROUGE F_1 results of large language models. ChatGPT version is gpt-3.5-turbo.

6.4. Large Language Models for Summarization

Large language models have exhibited impressive capabilities in zero- and few-shot settings, leading to the new paradigm of prompting. There are two application approaches of LLM, subscribing online API (like ChatGPT API (Ouyang et al., 2022)) or running locally (like Llama (Touvron et al., 2023a)). Both of them rely on computational resource support heavily, thus leaving positions for language models that are smaller but more expert.

To evaluate the zero-shot performance of the proposed method, those summarization tasks are tested on mainstream large language models for comparison. The implemented models are ChatGPT (gpt-3.5-turbo), Llama-2-7B (Touvron et al., 2023b), Llama-2-13B, Llama-30B, and our model with PROM. The prompt is "*Summarize the given document. Document: {doc} Summary:*". The results of ROUGE are shown in Table 7. It is observed that our model surpasses Llama models in the zero-shot setting. Compared to ChatGPT, our model achieves better scores on CNN/DM, NYT, and WikiHow, is comparable on BillSum and arXiv, and shows inferiority on Xsum.

We also conduct human-like evaluations using G-EVAL (Liu et al., 2023), which prompts ChatGPT to score the summaries. Zero-shot generations on CNN/DM and Xsum of PROM and Llama-13B are compared.[1] We evaluated the summaries of 100 samples from PROM and Llama-13B. The average score on CNN/DM is 3.53 for PROPM and 3.36 for Llama-13B, and the score on Xsum is 3.57 for PROM and 3.02 for Llama-13B. This suggests that continuous pre-training is an effective method for the zero-shot performance of a small expert model. This also proves that zero-shot small models with our method show a certain degree of competitiveness in the age of LLM.

6.5. Data Bias

This section analyzes experimental results considering dataset features.

Extractiveness & Abstractiveness Level. As is shown in Section 5.1, downstream datasets vary in the tendency to extract fragments from the source document. In Table 5, gains on BillSum and XSum are more moderate than other tasks. The possible reasons can be that (i) PROM breaks the continuous copying for phrase enhancement, which may disturb the extractions of BillSum. (ii) Xsum has a strong preference for abstraction and conciseness and thus benefits less from copying.

Lead Bias. The position distribution of the salient information is another essential feature of summarization tasks. Lead bias can be leveraged to make large margin improvements in the unsupervised setting (Yang et al., 2020a; Fabbri et al., 2021a). Figure 4 presents the position distributions of overlaps in the source documents. It shows that datasets have different levels of position preference. CNN/DM and NYT show stronger lead bias, while others have flatter curves.

Following Yang et al. (2020a) and Fabbri et al. (2021a), we consider a loose unsupervised setting, where the model can not see summary data but can leverage this domain feature. As mentioned in Section 4, we create the lead-biased data using the 29G subset and implement continuous pre-training. Zero-shot results in table 8 indicate that our method can achieve better scores with the lead bias. The results also prove that PROM can surpass TED and WikiTransfer methods under a similar setting and thus again verify the effectiveness.

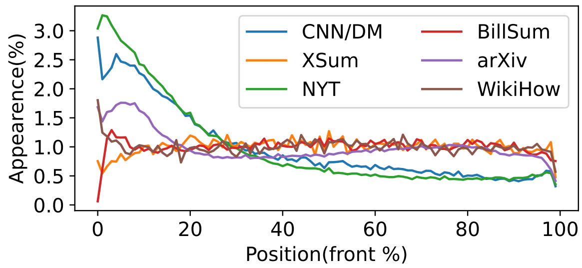

Figure 4: Position distributions of the overlaps across datasets.

Model	**R$_1$**	**R$_2$**	**R$_L$**
TED	38.38	16.49	35.08
WikiTransfer (w/o bin)	39.11	16.98	35.66
PROM	39.78	17.19	36.12

Table 8: Zero-shot ROUGE F_1 scores with lead bias on CNN/DM.

7. Conclusion

In this work, we propose PROM, a novel method to enhance phrase copying, which makes contributions to abstractive summarization in both supervised and zero-shot settings. This paper also gives a systematic study of copying methods for abstractive summarization. The proposed PROM encourages the copying of phrases and surpasses previous copying methods in fine-tuning. PROM is further utilized in pre-training and achieves improvements in zero-shot performance on a wide range of summarization tasks. The experimental results also prove that PROM shows advantages in faithfulness, entity coverage, and human evaluation. The scalability of PROM is shown across datasets that vary in genre, extractiveness level, and position bias. Our pre-trained model still has a practical significance in the zero-shot setting compared to large language models.

[1]The prompt line follows https://github.com/nlpyang/geval.

8. Bibliographical References

Bin Bi, Chenliang Li, Chen Wu, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2020. PALM: pre-training an autoencoding&autoregressive language model for context-conditioned generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8681–8691. Association for Computational Linguistics.

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

Arthur Brazinskas, Mirella Lapata, and Ivan Titov. 2020. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5151–5169. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana. Association for Computational Linguistics.

Xiuying Chen, Mingzhe Li, Xin Gao, and Xiangliang Zhang. 2022. Towards improving faithfulness in abstractive summarization. *arXiv preprint arXiv:2210.01877*.

Yiran Chen, Pengfei Liu, Ming Zhong, Zi-Yi Dou, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. CDEvalSumm: An empirical study of cross-dataset evaluation for neural summarization systems. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3679–3691, Online. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained language model representations for language generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4052–4059, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexander Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, Dragomir Radev, and Yashar Mehdad. 2021a. Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 704–717, Online. Association for Computational Linguistics.

Alexander R. Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq R. Joty, Dragomir R. Radev, and Yashar Mehdad. 2021b. Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 704–717. Association for Computational Linguistics.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4098–4109. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *CoRR*, abs/1911.12237.

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

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Pengcheng He, Baolin Peng, Liyang Lu, Song Wang, Jie Mei, Yang Liu, Ruochen Xu, Hany Hassan Awadalla, Yu Shi, Chenguang Zhu, Wayne Xiong, Michael Zeng, Jianfeng Gao, and Xuedong Huang. 2022. Z-code++: A pre-trained language model optimized for abstractive summarization. *CoRR*, abs/2208.09770.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A large scale chinese short text summarization dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1967–1972. The Association for Computational Linguistics.

Xinyu Hua and Lu Wang. 2017. A pilot study of domain adaptation effect for neural abstractive summarization. In *Proceedings of the Workshop on New Frontiers in Summarization, NFiS@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 100–106. Association for Computational Linguistics.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of reddit posts with multi-level memory networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2519–2531. Association for Computational Linguistics.

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd*

Workshop on New Frontiers in Summarization, pages 48–56, Hong Kong, China. Association for Computational Linguistics.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *CoRR*, abs/1810.09305.

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

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020b. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9332–9346. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020c. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9332–9346. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Haoran Li, Song Xu, Peng Yuan, Yujia Wang, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2021. Learn to copy from the copying history: Correlational copy network for abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4091–4101. Association for Computational Linguistics.

Xiaochen Liu, Yang Gao, Yu Bai, Jiawei Li, Yinan Hu, Heyan Huang, and Boxing Chen. 2022a. PSP: pre-trained soft prompts for few-shot abstractive summarization. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 6355–6368. International Committee on Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Yang Liu and Mirella Lapata. 2019. 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 (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022b. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Guulçehre, and Bing Xiang. 2016a. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016b. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290. ACL.

Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen McKeown, and Bing Xiang. 2021. Entity-level factual consistency of abstractive text summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2727–2733, Online. Association for Computational Linguistics.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020a. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2401–2410.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020b. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2401–2410. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020b. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5370–5381. Association for Computational Linguistics.

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

Horacio Saggion and Thierry Poibeau. 2013. Automatic text summarization: Past, present and future. In *Multi-source, multilingual information extraction and summarization*, pages 3–21. Springer.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

David Wan and Mohit Bansal. 2022. FactPEGASUS: Factuality-aware pre-training and fine-tuning for abstractive summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1010–1028, Seattle, United States. Association for Computational Linguistics.

Danqing Wang, Pengfei Liu, Ming Zhong, Jie Fu, Xipeng Qiu, and Xuanjing Huang. 2019a. Exploring domain shift in extractive text summarization. *arXiv preprint arXiv:1908.11664*.

Danqing Wang, Pengfei Liu, Ming Zhong, Jie Fu, Xipeng Qiu, and Xuanjing Huang. 2019b. Exploring domain shift in extractive text summarization. *CoRR*, abs/1908.11664.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Ziqiang Cao, Sujian Li, Hua Wu, and Haifeng Wang. 2021. BASS: Boosting abstractive summarization with unified semantic graph. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6052–6067, Online. Association for Computational Linguistics.

Wen Xiao and Giuseppe Carenini. 2022. Entity-based spancopy for abstractive summarization to improve the factual consistency. *CoRR*, abs/2209.03479.

Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1355–1362. Association for Computational Linguistics.

Ziyi Yang, Chenguang Zhu, Robert Gmyr, Michael Zeng, Xuedong Huang, and Eric Darve. 2020a. TED: A pretrained unsupervised summarization model with theme modeling and denoising. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1865–1874. Association for Computational Linguistics.

Ziyi Yang, Chenguang Zhu, Robert Gmyr, Michael Zeng, Xuedong Huang, and Eric Darve. 2020b. Ted: A pretrained unsupervised summarization model with theme modeling and denoising. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1865–1874.

Tiezheng Yu, Zihan Liu, and Pascale Fung. 2021. AdaptSum: Towards low-resource domain adaptation for abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5892–5904, Online. Association for Computational Linguistics.

Haopeng Zhang, Semih Yavuz, Wojciech Kryściński, Kazuma Hashimoto, and Yingbo Zhou. 2022. Improving the faithfulness of abstractive summarization via entity coverage control. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 528–535.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020b. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2204–2213. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020c. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Lulu Zhao, Fujia Zheng, Weihao Zeng, Keqing He, Weiran Xu, Huixing Jiang, Wei Wu, and Yanan Wu. 2022. Domain-oriented prefix-tuning: Towards efficient and generalizable fine-tuning for zero-shot dialogue summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4848–4862, Seattle, United States. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104, Vancouver, Canada. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2018. Sequential copying networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4987–4995. AAAI Press.

A. Example

Table 9 shows the example in Table 1 in details where only irrelevant sentences are omitted. We also present the overlapped bi-grams that are used for C in Eq. 3 and $\mathcal{L}_{copy}$ in Eq. 4.

The example shows the nature of the copied bi-grams to some extent. (i) Intersection with entities is usually contained, e.g., Hollywood actor, WikiLeaks founder, Ecuadorian Embassy. (ii) Some idiomatic expressions are contained, e.g., take shelter in, to do something, by doing something.

Article: Hollywood actor John Cusack is the latest supporter to visit WikiLeaks founder Julian Assange in his continued stay at the Ecuadorian Embassy... Assange has avoided being extradited to Sweden by taking shelter in the Ecuadorean Embassy in London since 2012... Hollywood actor John Cusack (pictured right) is the latest supporter to visit WikiLeaks founder Julian Assange (left) in his continued stay at the Ecuadorian Embassy - where he has remained since 2012... The Australian has been granted political asylum by Ecuador but faces arrest if he leaves the embassy... Mr Assange believes if he is sent to Sweden he will be extradited to the US, where he could face 35 years in prison for publishing on WikiLeaks classified documents related to US activities in Iraq and Afghanistan...
Summary: Hollywood actor is latest supporter to visit WikiLeaks founder Assange. Pictured arriving at the Ecuadorian Embassy where Assange is staying. Assange is avoiding extradition to Sweden by taking shelter in embassy.
bi-grams (Hollywood, actor), (latest, supporter), (supporter, to), (to, visit), (visit, WikiLeaks), (WikiLeaks, founder), (at, the), (the, Ecuador), (Ecuador, an), (an, Embassy), (to, Sweden), (Sweden, by), (by, taking), (taking, shelter), (shelter, in)

Table 9: An example for the copying indicator from CNN/DM.

B. Experimental Details

In empirical studies, our implementation is following *Transformers* library. The best scores are selected by the validation set and reported in the tables. All experiments are conducted on 32GB NVIDIA V100 GPUs. Our best setup of fine-tuning on CNN/DM is 64 for batch size, 5e-5 for learning rate, 30000 for step numbers. For pre-training, the better setup is 1e-6 for learning rate, 80 for batch size. In pre-training data construction, the parameters we use are as follows: the maximum sentence number for D_{chunk} is 8, the minimum is 4, min_{EFD} is 3, $m\%$ is 0.25, $\hat{m}$ is 3.

Since the implementations of ROUGE-L may differ between packages, we clarify that ROUGE-L scores that we report in our paper are from *rouge-scorer* package, which split texts using $\backslash n$. To avoid misunderstanding or confusion, we present another ROUGE-L that ignores $\backslash n$ in Table 10.

Model	**R_1**	**R_2**	**R_L**
PROM	44.47	21.59	31.06
PROM$_{two_stage}$	44.35	21.61	31.15
PROM$_{pre-train}$	44.59	21.66	31.37
Dataset	**R_1**	**R_2**	**R_L**
CNN/DM	37.87	15.91	24.93
XSum	22.96	6.05	17.27
NYT	36.95	17.21	25.67
WikiHow	25.90	6.37	16.17
BillSum	44.35	21.61	23.64
arXiv	40.05	14.66	19.93

Table 10: ROUGE F_1 scores. The upper part is fine-tuning results, and the lower part is zero-shot results.

Before the pre-training, a pilot experiment is conducted to compare PROM with the baseline model BART. They are trained on a mini pre-training dataset derived from a small subset of the corpora. Scores are shown in Table 11.

Model	**R_1**	**R_2**	**R_L**	**R_1**	**R_2**	**R_L**
	CNN/DM			**BillSum**		
BART	33.47	12.90	30.59	36.97	13.10	30.93
PROM$_{\text{mini}}$	34.34	13.49	31.32	37.56	12.97	31.53
	arXiv					
BART	33.25	9.8	28.82			
PROM$_{\text{mini}}$	34.23	9.69	29.84			

Table 11: Zero-shot ROUGE F_1 numbers of pilot experiments.

C. Human Evaluation

We present our implementation of human evaluation in this section. We first show the instruction that explains the experiment object and measurement, i.e., faithfulness, informativeness, and readability. Then, we show the scoring rules.

Instructions

We illustrate the features: faithfulness, informativeness, and readability.
1) **Faithfulness.** Faithfulness means factual consistency with the context. Please avoid using general knowledge, and only consider it in the context of the provided document. The summary is inconsistent if facts in the summary are not supported by the document. Two typical cases are conflict and hallucination.
(i) The summary contradicts the information in the document. The summary might say "A fire broke out in Seattle", but the document says it broke out in Portland. Or the summary might say "the Republicans won the election", but the document indicates the Democrats won instead.
(ii) The summary adds (hallucinates) a fact that is not mentioned anywhere in the document. For example, the summary might say that "A fire broke out at 2 am", but the document does not mention the time when the fire broke out.
2) **Informativeness.** It means that a summary expresses the main points of the document. A summary should contain relevant and important information and few unimportant details. If you select the summary to be not consistent with the document, please only consider the consistent information when evaluating this category.
3) **Readability.** The summaries are written by human or generated by language models. A summary is readable/fluent if free from language problems. A less readable summary is confusing and difficult to understand.
We present an example in Table 12. This article reports the accidental death of Alexys Brown. The main information is the accident and the appeal to raise money and minor points can be the investigation, postmortem examination, etc. Summary 1 can be reasonable and acceptable. Summary 2 misses a major point, the appeal, thus not informative. Both summary 3 and summary 4 show unfaithfulness. Summary 3 makes a factual mistake that Alexys died *of cancer*. This contradicts the article. Summary 4 adds a hallucination that Alexys is *three-year-old*. Summary 5 is confusing because grammar flaws impair readability.

Scoring rules

Please annotate which one of the 2 summaries is better in the four aspects separately. For example, if summary #1 is better than summary #2 in the aspect of informativeness, then type “1” behind ‘*****Informativeness:‘. It means summary #1 wins over summary #2 on informativeness. And if summary #2 wins in readability, then type “2” behind ‘*****Readability:‘. If the two summaries draw with each other (come out even) in an aspect, then type “0” in the cell below that aspect. There may be repeated summaries of an article, and please make sure that their scores are all 0.
Your results are 3 integers for each sample. The scores are in the order of *faithfulness, informativeness, readability*. The format is shown in Table 13.

Article: Alexys Brown, also known as Lexi, died at her home in Emmadale Close, Weymouth, on Thursday. An investigation is under way to discover how she became trapped. A post-mortem examination is due to be carried out this week. It was originally hoped the appeal would raise £2,000. Alison Record, who started the Just Giving appeal, said she was "heart broken" over the death. “Everybody by now has heard of the terrible tragedy the Brown family have suffered with the loss of their beautiful and beloved little girl Lexi,” the appeal page reads. Many other comments have been posted on the appeal page. Steph Harris said: “Thinking of you all at this devastating time, fly high beautiful princess. Love Steph and family xxx” Lesley Andrews added: “No amount of money will take away the pain, but so much love comes with every penny. Take care. xx” Aster Group, the housing association responsible for managing the home, is assisting with the police investigation. The Health and Safety Executive (HSE) is also investigating. Dorset County Council said it had not installed the disabled lift at the property.
Summary #1: An appeal to raise money for the family of a girl who died after getting stuck in a lift was originally hoped for raising 2,000 pounds.
Summary #2 (informativeness): Alexys Brown, also known as Lexi, died at her home in Emmadale Close, Weymouth, on Thursday.
Summary #3 (faithfulness): Alexys Brown, also known as Lexi, died <u>of cancer</u>. The appeal was originally hoped for raising 2,000 pounds.
Summary #4 (faithfulness): An appeal to raise money for Alexys Brown, a <u>three-year-old girl</u> who died after getting stuck in a lift was originally hoped for raising 2,000 pounds.
Summary #5 (readability): An appeal to raise the family of Alexys Brown became trapped in a lift would raise 2,000 pounds.

Table 12: An example for the instructions.

Article: Alexys Brown, also known as Lexi, died at her home in Emmadale Close, Weymouth, on Thursday. An investigation is under way to discover how she became trapped. A post-mortem examination is due to be carried out this week. It was originally hoped the appeal would raise £2,000. Alison Record, who started the Just Giving appeal, said she was "heart broken" over the death. “Everybody by now has heard of the terrible tragedy the Brown family have suffered with the loss of their beautiful and beloved little girl Lexi,” the appeal page reads. Many other comments have been posted on the appeal page. Steph Harris said: “Thinking of you all at this devastating time, fly high beautiful princess. Love Steph and family xxx” Lesley Andrews added: “No amount of money will take away the pain, but so much love comes with every penny. Take care. xx” Aster Group, the housing association responsible for managing the home, is assisting with the police investigation. The Health and Safety Executive (HSE) is also investigating. Dorset County Council said it had not installed the disabled lift at the property.
Summary #1: Alexys Brown, also known as Lexi, died at her home in Emmadale Close, Weymouth, on Thursday. **Summary #2:** Alexys Brown, also known as Lexi, died <u>of cancer</u>. The appeal was originally hoped for raising 2,000 pounds.
Scores: *****Faithfulness: 1 *****Informativeness: 2 *****Readability: 0

Table 13: An example for the scoring format.