PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization

Jingqing Zhang [*1] **Yao Zhao** [*2] **Mohammad Saleh** [2] **Peter J. Liu** [2]

Abstract

Recent work pre-training Transformers with self-supervised objectives on large text corpora has shown great success when fine-tuned on downstream NLP tasks including text summarization. However, pre-training objectives tailored for abstractive text summarization have not been explored. Furthermore there is a lack of systematic evaluation across diverse domains. In this work, we propose pre-training large Transformer-based encoder-decoder models on massive text corpora with a new self-supervised objective. In PEGASUS, important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to an extractive summary. We evaluated our best PEGASUS model on 12 downstream summarization tasks spanning news, science, stories, instructions, emails, patents, and legislative bills. Experiments demonstrate it achieves state-of-the-art performance on all 12 downstream datasets measured by ROUGE scores. Our model also shows surprising performance on low-resource summarization, surpassing previous state-of-the-art results on 6 datasets with only 1000 examples. Finally we validated our results using human evaluation and show that our model summaries achieve human performance on multiple datasets.

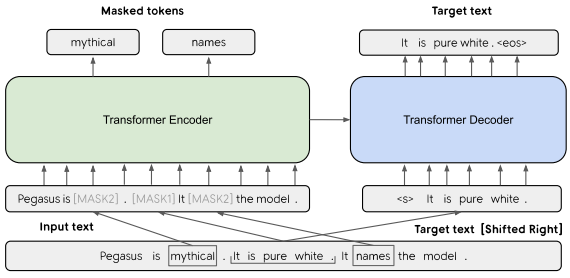


Figure 1: The base architecture of PEGASUS is a standard Transformer encoder-decoder. Both GSG and MLM are applied simultaneously to this example as pre-training objectives. Originally there are three sentences. One sentence is masked with `[MASK1]` and used as target generation text (GSG). The other two sentences remain in the input, but some tokens are randomly masked by `[MASK2]` (MLM).

1 Introduction

Text summarization aims at generating accurate and concise summaries from input document(s). In contrast to extractive summarization which merely copies informative fragments from the input, abstractive summarization may generate novel words. A good abstractive summary covers principal information in the input and is linguistically fluent.

In abstractive summarization, sequence-to-sequence (Sutskever et al., 2014) has become a dominant framework using encoder-decoder architectures based on RNNs (Chung et al., 2014; Hochreiter & Schmidhuber, 1997) and more recently Transformers (Vaswani et al., 2017). Most prior work on neural abstractive summarization relied on large-scale, high-quality datasets of supervised document-summary pairs (Hermann et al., 2015) and achieved promising results (Rush et al., 2015; Nallapati et al., 2016; See et al., 2017). In recent years, there has been increased interest in collecting new summarization datasets that have more abstractive summaries (Narayan et al., 2018), have longer documents, (Cohan et al., 2018; Sharma et al., 2019), utilize multiple documents (Fabbri et al., 2019), and are sourced from diverse domains (Grusky et al., 2018; Koupaee & Wang, 2018; Kim et al., 2019; Kornilova & Eidelman, 2019; Zhang & Tetreault, 2019);

[*]Equal contribution [1]Data Science Institute, Imperial College London, London, UK [2]Brain Team, Google Research, Mountain View, CA, USA. Correspondence to: Jingqing Zhang <jingqing.zhang15@imperial.ac.uk>, Yao Zhao <yaozhaoyz@google.com>, Mohammad Saleh <msaleh@google.com>, Peter J. Liu <peterjliu@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

however, there has been little work on systematic evaluation of models across these broad settings.

Contemporaneously, the adoption of Transformer models (Vaswani et al., 2017) pre-trained using self-supervised objectives on large text corpora (Radford et al., 2018a; Devlin et al., 2019) have improved performance on many NLP tasks (Wang et al., 2018; Rajpurkar et al., 2016).

Recent work leveraging such pre-training for Transformer-based sequence-to-sequence models (Dong et al., 2019; Song et al., 2019; Rothe et al., 2019; Lewis et al., 2019; Raffel et al., 2019) has extended the success to text generation, including abstractive summarization.

In this work, we study pre-training objectives specifically for abstractive text summarization and evaluate on 12 downstream datasets spanning news (Hermann et al., 2015; Narayan et al., 2018; Grusky et al., 2018; Rush et al., 2015; Fabbri et al., 2019), science (Cohan et al., 2018), short stories (Kim et al., 2019), instructions (Koupaee & Wang, 2018), emails (Zhang & Tetreault, 2019), patents (Sharma et al., 2019), and legislative bills (Kornilova & Eidelman, 2019). We find that masking whole sentences from a document and generating these gap-sentences from the rest of the document works well as a pre-training objective for downstream summarization tasks. In particular, choosing putatively important sentences outperforms lead or randomly selected ones. We hypothesize this objective is suitable for abstractive summarization as it closely resembles the downstream task, encouraging whole-document understanding and summary-like generation. We call this self-supervised objective Gap Sentences Generation (GSG). Using GSG to pre-train a Transformer encoder-decoder on large corpora of documents (Web and news articles) results in our method, Pre-training with Extracted Gap-sentences for Abstractive SUmmarization Sequence-to-sequence models, or *PEGASUS*.

With our best 568M parameter model trained on the recently introduced C4 (Raffel et al., 2019) corpus we equal or exceed state-of-the-art on the 12 summarization tasks we consider. We further push forward the state-of-the-art using a newly collected text corpus comprised of news-like articles we call HugeNews, including the highly competitive XSum and CNN/DailyMail summarization datasets.

Large-scale document-summary datasets are rare and in practice there is a mismatch between research datasets and real-world use-cases where collecting summaries is expensive; the most common setting is that of low-resource summarization. We simulate this setting and show that our model is able to adapt very quickly when fine-tuning with small numbers of supervised pairs, obtaining state-of-the-art results in 6 datasets with only 1000 examples.

Qualitatively we observed high quality outputs from our best models and validated this in human evaluation studies. We found that PEGASUS summaries are at least as good as reference summaries for the datasets we assessed – XSum, CNN/DailyMail, and Reddit TIFU – even at low-levels of supervision.

To summarize our contributions:

- We propose a new self-supervised pre-training objective for abstractive summarization, gap-sentences generation, and study strategies for selecting those sentences.
- We evaluate the proposed pre-training objective on a broad range of downstream summarization tasks, with careful ablations to choose the best model settings, which we use to train a 568M parameter PEGASUS model that surpasses or is on-par with the state-of-the-art on all 12 downstream datasets considered.
- We show how good abstractive summarization performance can be achieved across broad domains with very little supervision by fine-tuning the PEGASUS model and surpassing previous state-of-the-art results on many tasks with as little as 1000 examples.
- We conducted human evaluation studies to validate our experimental design and demonstrate human-level summarization performance on XSum, CNN/DailyMail, and Reddit TIFU.

2 Related Work

Dai & Le (2015); Ramachandran et al. (2017) used LM and autoencoder pre-training on in-domain data to improve performance of RNN sequence models. However, the combination of pre-training with much larger external text corpora (such as Wikipedia, books, or Web-pages) and Transformer-based sequence models has led to a dramatic improvement in performance when fine-tuned for both natural language understanding and text generation tasks (Radford et al., 2018a; Devlin et al., 2019; Rothe et al., 2019; Yang et al., 2019; Joshi et al., 2019; Song et al., 2019; Dong et al., 2019; Lewis et al., 2019). Most similar to our approach are Transformer encoder-decoder models pre-trained on some masked input pre-training objective.

MASS (Song et al., 2019) proposed masked sequence-to-sequence generation that reconstructs a sentence fragment given the remaining part of the sentence. A single sentence fragment was randomly selected.

UniLM (Dong et al., 2019) proposed jointly training on three types of language modeling tasks: unidirectional (left-to-right and right-to-left), bidirectional (word-level mask,

with next sentence prediction), and sequence-to-sequence (word-level mask) prediction.

T5 (Raffel et al., 2019) generalized the text-to-text framework to a variety of NLP tasks and showed the advantage of scaling up model size (to 11 billion parameters) and pre-training corpus, introducing C4, a massive text corpus derived from Common Crawl, which we also use in some of our models. T5 was pre-trained with randomly corrupted text spans of varying mask ratios and sizes of spans.

BART (Lewis et al., 2019) introduced a denoising autoencoder to pre-train sequence-to-sequence models. BART corrupted text with an arbitrary noising function and learned to reconstruct the original text. For generation tasks, the noising function was text infilling which used single mask tokens to mask random sampled spans of text.

In contrast to MASS, UniLM, BART and T5, the proposed PEGASUS masks multiple whole sentences rather than smaller continuous text spans. In our final objective we deterministically choose sentences based on importance, rather than randomly. As in T5, PEGASUS does not reconstruct full input sequences, and only generates the masked sentences as a single output sequence. In this work we focus entirely on downstream summarization (generative) tasks and do not evaluate on NLU classification tasks.

There has been some work on the low-resource, summarization setting using the CNN/DailyMail dataset. Radford et al. (2018b) showed that a large Transformer language model pre-trained on Web text could generate summaries if prompted with "TL;DR", achieving a ROUGE-2 of 8.27 on CNN/DailyMail. Khandelwal et al. (2019) pre-trained a Transformer language model on Wikipedia, and fine-tuned using 3000 examples, achieving 13.1 ROUGE-2.

3 Pre-training Objectives

We propose a new pre-training objective, GSG, in this work, but for comparison, we also evaluate BERT's masked-language model objective, in isolation and in conjunction with GSG.

3.1 Gap Sentences Generation (GSG)

We hypothesize that using a pre-training objective that more closely resembles the downstream task leads to better and faster fine-tuning performance. Given our intended use for abstractive summarization, our proposed pre-training objective involves generating summary-like text from an input document. In order to leverage massive text corpora for pre-training, we design a sequence-to-sequence self-supervised objective in the absence of abstactive summaries. A naive option would be to pre-train as an extractive summarizer; however, such a procedure would only train a model to copy sentences, thus not suitable for abstractive summarization.

Inspired by recent success in masking words and contiguous spans (Joshi et al., 2019; Raffel et al., 2019), we select and mask whole sentences from documents, and concatenate the gap-sentences into a pseudo-summary. The corresponding position of each selected gap sentence is replaced by a mask token `[MASK1]` to inform the model. *Gap sentences ratio*, or *GSR*, refers to the number of selected gap sentences to the total number of sentences in the document, which is similar to *mask rate* in other works.

To even more closely approximate a summary, we select sentences that appear to be important/principal to the document. The resulting objective has both the empirically demonstrated benefits of masking, and anticipates the form of the downstream task.

We consider 3 primary strategies for selecting m gap sentences without replacement from a document, $D = \{x_i\}_n$, comprised of n sentences:

Random Uniformly select m sentences at random.

Lead Select the first m sentences.

Principal Select top-m scored sentences according to importance. As a proxy for importance we compute ROUGE1-F1 (Lin, 2004) between the sentence and the rest of the document, $s_i = rouge(x_i, D \setminus \{x_i\}), \forall i$.

In this formulation sentences are scored independently (**Ind**) and the top m selected. We also consider selecting them sequentially (**Seq**) as in Nallapati et al. (2017) by greedily maximizing the ROUGE1-F1 between selected sentences, $S \cup \{x_i\}$, and remaining sentences, $D \setminus (S \cup \{x_i\})$ as in Algorithm 1.

Algorithm 1 Sequential Sentence Selection

$S := \emptyset$
for $j \leftarrow 1$ to m **do**
  $s_i := rouge\Big(S \cup \{x_i\}, D \setminus (S \cup \{x_i\})\Big)$
  $\forall i \;\; s.t. \;\; x_i \notin S$
  $k := \arg\max_i \{s_i\}_n$
  $S := S \cup \{x_k\}$
end for

When calculating ROUGE1-F1, we also consider n-grams as a set (**Uniq**) instead of double-counting identical n-grams as in the original implementation (**Orig**). This results in four variants of the principal sentence selection strategy, choosing Ind/Seq and Orig/Uniq options.

An example containing lead, random and principal gap sentence selection strategies are shown in Figure 2.

INVITATION ONLY We are very excited to be co-hosting a major drinks reception with our friends at Progress. This event will sell out, so make sure to register at the link above. Speakers include Rajesh Agrawal, the London Deputy Mayor for Business, Alison McGovern, the Chair of Progress, and Seema Malhotra MP. Huge thanks to the our friends at the ACCA, who have supported this event. The Labour Business Fringe at this year's Labour Annual Conference is being co-sponsored by Labour in the City and the Industry Forum. Speakers include John McDonnell, Shadow Chancellor, and Rebecca Long-Bailey, the Shadow Chief Secretary to the Treasury, and our own Chair, Kitty Ussher. Attendance is free, and refreshments will be provided.

Figure 2: An example of sentences (from the C4 corpus) selected by Random, Lead and Ind-Orig respectively. Best viewed in color.

3.2 Masked Language Model (MLM)

Following BERT, we select 15% tokens in the input text, and the selected tokens are (1) 80% of time replaced by a mask token `[MASK2]`, or (2) 10% of time replaced by a random token, or (3) 10% of time unchanged. We apply MLM to train the Transformer encoder as the sole pre-training objective or along with GSG. When MLM is the sole pre-training objective, the Transformer decoder shares all parameters with encoder when fine-tuning on downstream tasks following Rothe et al. (2019).

Figure 1 simultaneously shows how both GSG and MLM are applied to the same example when used in conjunction. However, we found that MLM does not improve downstream tasks at large number of pre-training steps (section 6.1.2), and chose not to include MLM in the final model $\text{PEGASUS}_{\text{LARGE}}$ (section 6.2).

4 Pre-training Corpus

For pre-training we considered two large text corpora:

- **C4**, or the Colossal and Cleaned version of Common Crawl, introduced in Raffel et al. (2019); consists of text from 350M Web-pages (750GB).

- **HugeNews**, a dataset of 1.5B articles (3.8TB) collected from news and news-like websites from 2013-2019. A whitelist of domains ranging from high-quality news publishers to lower-quality sites such as high-school newspapers, and blogs was curated and used to seed a web-crawler. Heuristics were used to identify news-like articles, and only the main article text was extracted as plain text.

5 Downstream Tasks/Datasets

For downstream summarization, we only used public abstractive summarization datasets, and access them through TensorFlow Summarization Datasets [1], which provides publicly reproducible code for dataset processing and train/validation/test splits. We used train/validation/test ratio of 80/10/10 if no split was provided, and 10% train split as validation if there was no validation split.

XSum (Narayan et al., 2018) consists of 227k BBC articles from 2010 to 2017 covering a wide variety of subjects along with professionally written single-sentence summaries.

CNN/DailyMail (Hermann et al., 2015) dataset contains 93k articles from the CNN, and 220k articles the Daily Mail newspapers. Both publishers supplement their articles with bullet point summaries. We use the non-anonymized variant used in See et al. (2017).

NEWSROOM (Grusky et al., 2018) is a large dataset containing 1.3M article-summary pairs written by authors and editors in the newsrooms of 38 major publications between 1998 and 2017.

Multi-News (Fabbri et al., 2019) is a multi-document summarization dataset consisting of 56k pairs of news articles and their human-written summaries from the site newser.com.

Gigaword (Rush et al., 2015) contains 4M examples extracted from news articles (seven publishers) from the Gigaword corpus (Graff et al., 2003). The task is to generate the headline from the first sentence.

arXiv, PubMed (Cohan et al., 2018) are two long document datasets of scientific publications from arXiv.org (113k) and PubMed (215k). The task is to generate the abstract from the paper body.

BIGPATENT (Sharma et al., 2019) consists of 1.3 million U.S. patents along with human summaries under nine patent classification categories.

WikiHow (Koupaee & Wang, 2018) is a large-scale dataset of instructions from the online WikiHow.com website. Each of 200k examples consists of multiple instruction-step paragraphs along with a summarizing sentence. The task is to generate the concatenated summary-sentences from the paragraphs.

Reddit TIFU (Kim et al., 2019) contains 120K posts of informal stories from the online discussion forum Reddit, more specifically the TIFU sub-reddit from 2013-Jan to 2018-Mar. The sub-reddit posts strictly follow the rule of writing a descriptive "TL;DR" summary and has higher qual-

[1] `https://www.tensorflow.org/datasets/catalog/overview`

ity than (Völske et al., 2017) (which used more subreddits) based on our manual inspection. We uses the TIFU-long subset (using TLDR as summaries) in the work.

AESLC (Zhang & Tetreault, 2019) consists of 18k email bodies and their subjects from the Enron corpus (Klimt & Yang, 2004), a collection of email messages of employees in the Enron Corporation.

BillSum (Kornilova & Eidelman, 2019) contains 23k US Congressional bills and human-written reference summaries from the 103rd-115th (1993-2018) sessions of Congress. We do not use the California test set which is out-of-distribution.

Following Grusky et al., the number of examples and extractive fragment coverage/density for all downstream datasets is illustrated in Appendix A.

6 Experiments

In a similar strategy to Raffel et al. (2019), to save time and computation we conducted pre-training ablation experiments using a reduced-size model with 223M parameters, **PEGASUS$_{\text{BASE}}$**, smaller batch size, and only 4 of 12 datasets before scaling up pre-training with the best settings to the final 568M parameters, **PEGASUS$_{\text{LARGE}}$**. The datasets (XSum, CNN/DailyMail, WikiHow and Reddit TIFU) were chosen for diversity in abstractiveness, writing style, and size.

PEGASUS$_{\text{BASE}}$ had $L = 12, H = 768, F = 3072, A = 12$ and PEGASUS$_{\text{LARGE}}$ had $L = 16, H = 1024, F = 4096, A = 16$, where L denotes the number of layers for encoder and decoder (i.e. Transformer blocks), H for the hidden size, F for the feed-forward layer size and A for the number of self-attention heads. We pre-trained PEGASUS$_{\text{BASE}}$ with a batch size of 256 and PEGASUS$_{\text{LARGE}}$ with a batch size of 8192. We refer to PEGASUS$_{\text{BASE}}$ without pre-training as **Transformer$_{\text{BASE}}$**.

We used sinusoidal positional encoding following Vaswani et al. (2017). For optimization, both pre-training and fine-tuning used Adafactor (Shazeer & Stern, 2018) with square root learning rate decay and dropout rate of 0.1.

We used greedy-decoding for studies in Section 6.1, and used beam-search with a length-penalty, α, as in Wu et al. (2016) for the final large model.

All experiments' hyper parameters can be found in Appendix C and reported numbers are in Appendix D and E.

6.1 Ablations on PEGASUS$_{\text{BASE}}$

We used PEGASUS$_{\text{BASE}}$ to evaluate choices of pre-training corpus, pre-training objective, and vocabulary size. For reproducibility, we evaluated the latter two using the publicly available C4 corpus.

Note that the y-axis in Figures 3, 4, 5 are normalized by the left-most bar using $\frac{1}{3}(\frac{R1}{R1_{\text{base}}} + \frac{R2}{R2_{\text{base}}} + \frac{RL}{RL_{\text{base}}})$ where $R1$, $R2$, RL are ROUGE F1 scores and $R1_{\text{base}}$, $R2_{\text{base}}$, RL_{base} are the scores of the configuration corresponding to the first bar.

With more pre-training steps, the model observed more documents in the pre-training corpus. A PEGASUS$_{\text{BASE}}$ model trained for 500k (highest we tried) steps did not observe all training examples on C4 nor HugeNews. Appendix B shows the number of pre-training steps had an unsurprisingly positive impact on downstream dataset performance. We used 500k steps for the ablation studies and the large model.

6.1.1 Pre-training Corpus

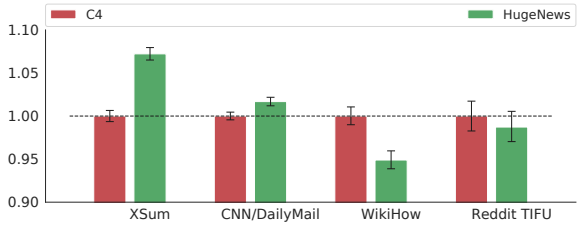


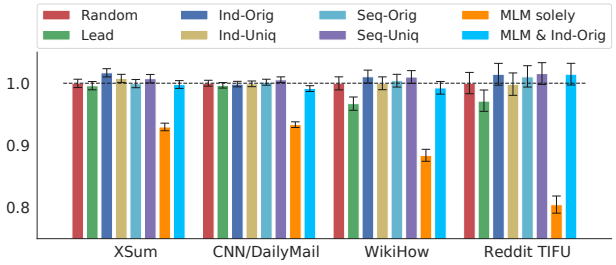
Figure 3: Effect of pre-training corpus. PEGASUS$_{\text{BASE}}$ pre-trained on C4 (350M Web-pages) and HugeNews (1.5B news-like documents).

Figure 3 shows that pre-training on HugeNews was more effective than C4 on the two news downstream datasets, while the non-news informal datasets (WikiHow and Reddit TIFU) prefer the pre-training on C4. This suggests pre-training models transfer more effectively to downstream tasks when their domains are aligned better.

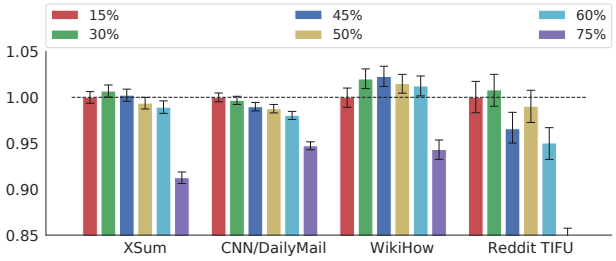
6.1.2 Effect of Pre-training Objectives

GSG We compared six variants of GSG (Lead, Random, Ind-Orig, Ind-Uniq, Seq-Orig, Seq-Uniq) while choosing 30% sentences as gap sentences. As shown in Figure 4a, Ind-Orig achieved the best performance followed by Seq-Uniq. Ind-Orig and Seq-Uniq were consistently better (or similar) than Random and Lead across the four downstream datasets. Lead had decent performance on the two news datasets but was significantly worse on the two non-news datasets, which agrees findings of lead bias in news datasets (See et al., 2017; Zhong et al., 2019). The results suggest choosing principal sentences works best for downstream summarization tasks, and we chose Ind-Orig for the PEGASUS$_{\text{LARGE}}$.

A significant hyper-parameter in GSG is the gap-sentences ratio (GSR). A low GSR makes the pre-training less challenging and computationally efficient. On the other hand, choosing gap sentences at a high GSR loses contextual in-



(a) Effect of pre-training objectives (30% GSR).



(b) Effect of gap sentences ratio with GSG (Ind-Orig).

Figure 4: Effect of pre-training settings with PEGASUS$_{\text{BASE}}$ pre-trained on C4.

formation necessary to guide the generation. We compared GSRs from 15% to 75%. For a fair comparison, the original documents were truncated to have up to 400 words. The *maximum input length*, L_{input} in the encoder and the *maximum target length*, L_{target} in the decoder were set as 512 tokens.

Figure 4b shows that different downstream datasets had slightly different optima. The best performance always had GSR lower than 50%. The model with 15% gap sentences achieved the highest ROUGE scores on CNN/DailyMail, while XSum/Reddit TIFU and WikiHow did better with 30% and 45% respectively. When scaling up to PEGASUS$_{\text{LARGE}}$ (Section 6.2), we chose an effective GSR of 30%.

MLM As mentioned, the MLM objective can either be applied solely or together with GSG. We jointly trained MLM with GSG Ind-Orig (MLM & Ind-Orig), which masks 30% sentences and extra 15% tokens in unselected sentences, as shown in Figure 1. Figure 4a shows that the model pre-trained with MLM alone performed significantly worse and MLM & Ind-Orig had similar performance as Random. Interestingly, when comparing MLM & Ind-Orig to Ind-Orig, we empirically observed MLM improved fine-tuning performance at early pre-training checkpoints (100k - 200k steps), but inhibited further gains with more pre-training steps (500k). Therefore, we chose not to include MLM in PEGASUS$_{\text{LARGE}}$.

6.1.3 Effect of Vocabulary

We compared two tokenization methods[2]: Byte-pair-encoding algorithm (**BPE**) (Wu et al., 2016; Sennrich et al., 2016), and SentencePiece Unigram algorithm (**Unigram**) proposed in Kudo (2018). We evaluated Unigram with different vocabulary sizes ranging from 32k to 256k. In these experiments, models were pre-trained for 500k steps on the C4 corpus with the Ind-Orig objective and 15% GSR. As shown in Figure 5, BPE and Unigram were comparable on news datasets while Unigram outperformed BPE

[2]Implemented in https://github.com/google/sentencepiece

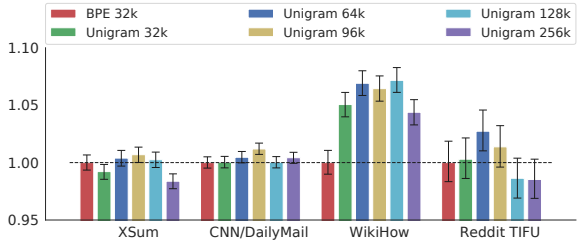


Figure 5: Effect of vocabulary with PEGASUS$_{\text{BASE}}$ trained on C4 (15% GSR, Ind-Orig).

on non-news datasets, especially WikiHow. On XSum and CNN/DailyMail, Unigram 96k achieved the highest ROUGE scores. On WikiHow and Reddit TIFU, the best configurations were Unigram 128k and 64k respectively. Therefore, we used the overall best vocabulary option Unigram 96k in PEGASUS$_{\text{LARGE}}$.

6.2 Larger Model Results

Compared with PEGASUS$_{\text{BASE}}$, the large model PEGASUS$_{\text{LARGE}}$ had increased capacity from larger hidden size (H : 768 $\rightarrow$ 1024, F : 3072 $\rightarrow$ 4096, A : 12 $\rightarrow$ 16), number of layers (L : 12 $\rightarrow$ 16) and traversed much more data, due to larger batch size (B : 256 $\rightarrow$ 8192) (same number of pre-training steps, 500k). We adopted the best practices found in the PEGASUS$_{\text{BASE}}$ ablation studies using the GSG (Ind-Orig) pre-training objective without MLM and Unigram vocabulary size of 96k. In total, PEGASUS$_{\text{LARGE}}$ had 568M parameters.

To encourage the model to copy, which is an important aspect of the more extractive datasets, we left 20% of selected sentences unchanged in the input instead of replacing with `[MASK1]`. We increased the GSR to 45% to achieve a similar number of "gaps" as the optimal 30% found above. We reported the performance of the models pre-trained on HugeNews and C4 separately. We conducted a simple hyper-parameter sweep of learning rate and length penalty,

Table 1: Results of $\text{PEGASUS}_{\text{LARGE}}$ and $\text{PEGASUS}_{\text{BASE}}$ on all downstream datasets compared with the previous SOTA, which are fetched from (Lewis et al., 2019; Shi et al., 2019; Fabbri et al., 2019; Koupaee & Wang, 2018; Kim et al., 2019; Subramanian et al., 2019; Song et al., 2019; Zhang & Tetreault, 2019; Kornilova & Eidelman, 2019). We only compared with previous abstractive models except on BillSum which had extractive results only. BIGPATENT, arXiv, PubMed and Multi-News datasets contain very long summaries and we truncate them to 256 tokens, in similar range compared to (Sharma et al., 2019; Cohan et al., 2018; Fabbri et al., 2019; Goodman et al., 2019). Best ROUGE numbers on each dataset and numbers within 0.15 of the best numbers are bolded.

R1/R2/RL	Dataset size	$\text{Transformer}_{\text{BASE}}$	$\text{PEGASUS}_{\text{BASE}}$	Previous SOTA	$\text{PEGASUS}_{\text{LARGE}}$ (C4)	$\text{PEGASUS}_{\text{LARGE}}$ (HugeNews)
XSum	226k	30.83/10.83/24.41	39.79/16.58/31.70	45.14/22.27/37.25	45.20/22.06/36.99	**47.21/24.56/39.25**
CNN/DailyMail	311k	38.27/15.03/35.48	41.79/18.81/38.93	**44.16**/21.28/40.90	43.90/21.20/40.76	**44.17/21.47/41.11**
NEWSROOM	1212k	40.28/27.93/36.52	42.38/30.06/38.52	39.91/28.38/36.87	**45.07/33.39/41.28**	**45.15/33.51/41.33**
Multi-News	56k	34.36/5.42/15.75	42.24/13.27/21.44	43.47/14.89/17.41	46.74/17.95/24.26	**47.52/18.72/24.91**
Gigaword	3995k	35.70/16.75/32.83	36.91/17.66/34.08	**39.14/19.92/36.57**	38.75/**19.96**/36.14	**39.12/19.86**/36.24
WikiHow	168k	32.48/10.53/23.86	36.58/15.64/30.01	28.53/9.23/26.54	**43.06/19.71/34.80**	41.35/18.51/33.42
Reddit TIFU	42k	15.89/1.94/12.22	24.36/6.09/18.75	19.0/3.7/15.1	**26.54/8.94/21.64**	**26.63/9.01/21.60**
BIGPATENT	1341k	42.98/20.51/31.87	43.55/20.43/31.80	37.52/10.63/22.79	**53.63/33.16/42.25**	53.41/32.89/42.07
arXiv	215k	35.63/7.95/20.00	34.81/10.16/22.50	41.59/14.26/23.55	**44.70/17.27/25.80**	**44.67/17.18/25.73**
PubMed	133k	33.94/7.43/19.02	39.98/15.15/25.23	40.59/15.59/23.59	**45.49/19.90/27.69**	45.09/19.56/27.42
AESLC	18k	15.04/7.39/14.93	34.85/18.94/34.10	23.67/10.29/23.44	**37.69/21.85/36.84**	37.40/21.22/36.45
BillSum	24k	44.05/21.30/30.98	51.42/29.68/37.78	40.80/23.83/33.73	**57.20**/39.56/**45.80**	**57.31/40.19/45.82**

Table 2: A comparison of $\text{PEGASUS}_{\text{LARGE}}$ with other pretrained models on XSum, CNN/DailyMail and Gigaword. Best ROUGE numbers and numbers within 0.15 of the best numbers are bolded.

R1/R2/RL	XSum	CNN/DailyMail	Gigaword
BERTShare (Rothe et al., 2019)	38.52/16.12/31.13	39.25/18.09/36.45	38.13/19.81/35.62
MASS (Song et al., 2019)	39.75/17.24/31.95	42.12/19.50/39.01	38.73/19.71/35.96
UniLM (Dong et al., 2019)	-	43.33/20.21/40.51	38.45/19.45/35.75
BART (Lewis et al., 2019)	45.14/22.27/37.25	**44.16**/21.28/40.90	-
T5 (Raffel et al., 2019)	-	43.52/**21.55**/40.69	-
$\text{PEGASUS}_{\text{LARGE}}$ (C4)	45.20/22.06/36.99	43.90/21.20/40.76	38.75/**19.96/36.14**
$\text{PEGASUS}_{\text{LARGE}}$ (HugeNews)	**47.21/24.56/39.25**	**44.17/21.47/41.11**	**39.12/19.86/36.24**

α, when fine-tuning $\text{PEGASUS}_{\text{LARGE}}$ on each downstream dataset.

CNN/DailyMail, Multi-News, arXiv, PubMed, BIGPATENT datasets contain input documents longer than the maximum input length ($L_{input} = 512$ tokens) in pre-training. This would present a problem for position embeddings which would never be updated for longer input lengths, but we confirm the postulation that sinusoidal positional encodings (Vaswani et al., 2017) generalize well when fine-tuning $\text{PEGASUS}_{\text{LARGE}}$ beyond the input lengths observed in training up to $L_{input} = 1024$ tokens. Since average input length in BIGPATENT, arXiv, PubMed and Multi-News are well beyond 1024 tokens, further scaling up L_{input} or applying a two-stage approach (Liu et al., 2018) may improve performance even more, although this is outside the scope of this work.

Tables 1 and 2 show the performance improvements of $\text{PEGASUS}_{\text{BASE}}$ and $\text{PEGASUS}_{\text{LARGE}}$ on downstream datasets. While $\text{PEGASUS}_{\text{BASE}}$ exceeded current state-of-the-art on many datasets, $\text{PEGASUS}_{\text{LARGE}}$ achieved better than state-of-the-art results on all downstream datasets using HugeNews, although C4 performed better on WikiHow.

The improvement from a Transformer model without pre-training ($\text{Transformer}_{\text{BASE}}$) to $\text{PEGASUS}_{\text{LARGE}}$ was more significant on smaller datasets. For example, the ROUGE2-F1 scores nearly tripled on AESLC and quintupled on Reddit TIFU. The large jumps in performance suggest that small text summarization datasets benefit the most from pre-training. We further investigate low resource summarization in Section 6.3.

6.3 Zero and Low-Resource Summarization

In real-world practice, it is often difficult to collect a large number of supervised examples to train or fine-tune a summarization model. To simulate the low-resource summarization setting, we picked the first 10^k ($k = 1, 2, 3, 4$) training examples from each dataset to fine-tune $\text{PEGASUS}_{\text{LARGE}}$ (HugeNews) . We fine-tuned the models up to 2000 steps with batch size 256, learning rate 0.0005, and picked the checkpoint with best validation performance. In Figure. 6, in 8 out of 12 datasets, with just 100 examples

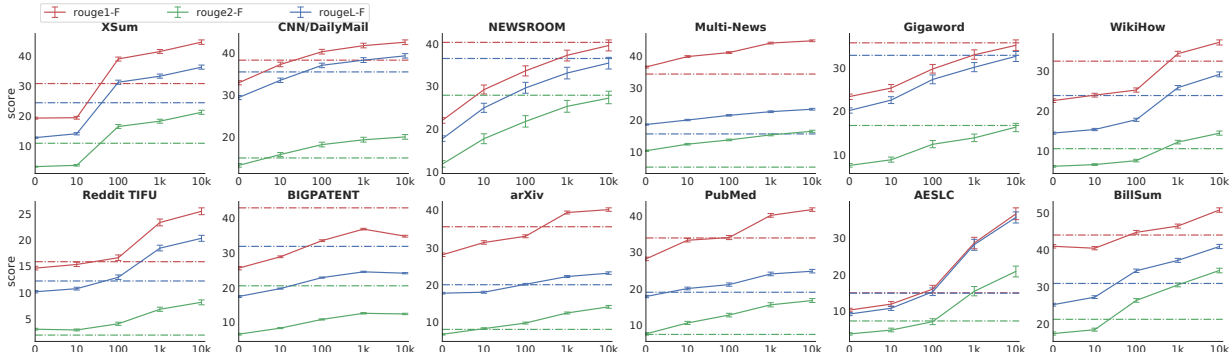


Figure 6: Fine-tuning with limited supervised examples. The solid lines are PEGASUS$_{\text{LARGE}}$ fine-tuned on 0 (zero shot), 10, 100, 1k,10k examples. The dashed lines are Transformer$_{\text{BASE}}$ models, equivalent in capacity as PEGASUS$_{\text{BASE}}$ and trained using the full supervised datasets, but with no pre-training. All numbers are reported in Appendix E.

Table 3: Human evaluation side-by-side results on Likert (1-5) scale (higher is better). Scores are bolded if they are not worse than human-level performance by $p < 0.01$.

Datasets	**XSum** mean (p-value)	**CNN/DailyMail** mean (p-value)	**Reddit TIFU** mean (p-value)
Experiment 1: pretrain comparison			
Human-written	3.0 (-)	3.1 (-)	3.2 (-)
PEGASUS$_{\text{LARGE}}$ (HugeNews)	**3.0** (0.6)	**3.6** (0.0001)	**3.2** (0.7)
PEGASUS$_{\text{LARGE}}$ (C4)	**3.1** (0.7)	**3.5** (0.009)	**3.1** (0.3)
Transformer$_{\text{BASE}}$	2.0 (3e-10)	**2.9** (0.06)	1.4 (5e-23)
Experiment 2: low resource			
Human-written	3.2 (-)	3.2(-)	3.3 (-)
PEGASUS$_{\text{LARGE}}$ (HugeNews) 10 examples	**2.8** (0.1)	**3.4** (0.007)	2.6 (0.006)
PEGASUS$_{\text{LARGE}}$ (HugeNews) 100 examples	**3.2** (0.5)	**3.4** (0.08)	2.1 (4e-8)
PEGASUS$_{\text{LARGE}}$ (HugeNews) 1000 examples	**3.4** (0.3)	**3.6** (0.07)	2.7 (0.01)
PEGASUS$_{\text{LARGE}}$ (HugeNews) full supervision	**3.4** (0.3)	**3.3** (0.1)	**2.8** (0.05)

PEGASUS$_{\text{LARGE}}$ could be fine-tuned to generate summaries at comparable quality to Transformer$_{\text{BASE}}$ trained on the full supervised datasets ranging from 20k to 200k examples. PEGASUS$_{\text{LARGE}}$ also beat previous state-of-the-art results on 6 out of 12 datasets with only 1000 fine-tuning examples.

On CNN/DailyMail, with half the number of parameters PEGASUS$_{\text{LARGE}}$ demonstrated much better zero-shot (ROUGE2-F=13.28) performance than GPT-2 (ROUGE2-F=8.27). Using only 1000 examples, PEGASUS$_{\text{LARGE}}$ achieved ROUGE2-F of 19.35, much higher than the 13.1 obtained in Khandelwal et al. (2019) with 3000 examples.

6.4 Qualitative Observations and Human Evaluation

Overall, we observed high-linguistic quality (in terms of fluency and coherence), closely emulating the style of ground-truth summaries. While some previous work suggested that maximum likelihood training results in repetitive text in model outputs (Welleck et al., 2019) we found this to be rare in our outputs and did not require additional counter-measures to mitigate dis-fluencies.

Although ROUGE clearly has its draw-backs (Kryscinski et al., 2019), over-penalizing abstractive approaches compared to extractive ones and having no sense of linguistic quality, we found that choosing perplexity-optimized models using aggregated ROUGE (rather than directly optimizing ROUGE as in Paulus et al. (2017)) resulted in qualitatively good models. Randomly sampled (by a program) model decodes across all datasets and a broad range of ROUGE scores can be found in Appendix I.We found that even low-ROUGE model summaries often were high-quality, Figure G.1.

To assess how close PEGASUS$_{\text{LARGE}}$ is to human performance we conducted human evaluation experiments on Amazon Mechanical Turk comparing model summaries with (human) reference summaries given the input document. The examples were drawn from the XSum, CNN/DailyMail, and Reddit TIFU datasets; the first two were chosen due to their popularity in past work, and the third was chosen for its significant difference in style. Workers were asked to rate the summaries on a 1-5 scale, with higher being better (full experiment details provided in Appendix F) and a paired t-test was used to assess whether scores were significantly different from human.

In the first experiment, PEGASUS$_{\text{LARGE}}$ (HugeNews), PEGASUS$_{\text{LARGE}}$ (C4), and Transformer$_{\text{BASE}}$ were compared with reference summaries; in the second experiment, PEGASUS$_{\text{LARGE}}$ (HugeNews) fine-tuned using 10, 100, 1000, and all supervised examples were compared with references; the results are shown in Table 3. According to the significance level of $p < 0.01$, both PEGASUS$_{\text{LARGE}}$ (HugeNews) and PEGASUS$_{\text{LARGE}}$ (C4) outputs were at least as good as the reference summaries in all cases. Even at low-levels of supervision PEGASUS$_{\text{LARGE}}$ (HugeNews) was not measurably worse than human summaries on XSum and CNN/DailyMail. In the Reddit TIFU case, however, perhaps due to its diverse writing styles, human performance required full supervision.

6.5 Test-set Overlap with Pre-training Corpus

The pre-training corpora are a large collection of documents from the Internet and potentially have overlap with the downstream test sets. In this section, we measured the extent of overlap between the pre-training corpus and downstream datasets. We also studied if the pre-trained model was able to exploit memorization to achieve higher performance on the downstream datasets.

To measure the overlap, we calculated similarities between all pairs of downstream test set targets and pre-training documents. We use the ROUGE-2 recall as a similarity measure (common 2-grams / test set targets 2-grams). It is not necessarily exact match even if the similarity score is 1.0. We filtered all test set examples that have similarity to any pre-training example above a threshold, and recalculated the ROUGE scores on the remaining test set. In Figure 7, we conducted this study on the pre-training corpus C4 and test set of XSum, CNN/Dailymail, Reddit TIFU and WikiHow, with a similarity threshold of 1.0 and 0.8. Results show that only XSum has significant amount of overlap 15% to 20%, and filtering those examples does not change ROUGE scores more than 1%. We also manually examined those overlapped examples with similarity of 1.0, and found that the models produce very different summaries compared to the human written ones, suggesting that there was no clear memorization.

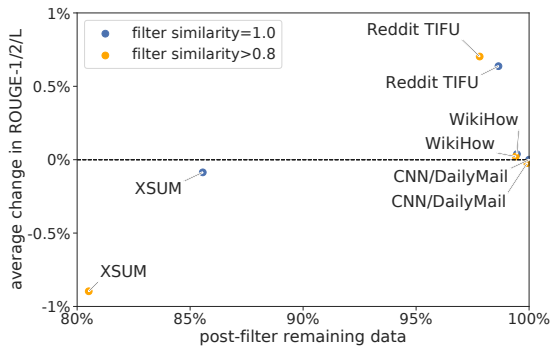


Figure 7: Percentage of overlap between C4 and downstream test sets, and ROUGE score changes after removing those overlapped examples in test sets.

6.6 Additional PEGASUS$_{\text{LARGE}}$ Improvements

Following our experiments on PEGASUS$_{\text{LARGE}}$ pre-trained on C4 and HugeNews, we pre-trained a PEGASUS$_{\text{LARGE}}$ model on both corpora and stochastically sampled important sentences. The PEGASUS$_{\text{LARGE}}$ (mixed,stochastic) model includes the changes: (1) The model was pre-trained on the mixture of C4 and HugeNews weighted by their number of examples. (2) The model dynamically chose gap sentences ratio uniformly between 15%-45%. (3) Importance sentences were stochastically sampled with 20% uniform noise on their scores. (4) The model was pre-trained for 1.5M steps instead of 500k steps, as we observed slower convergence of pre-training perplexity. (5) The SentencePiece tokenizer was updated to encode the newline character. The PEGASUS$_{\text{LARGE}}$ (mixed, stochastic) model achieved best results on almost all downstream tasks, as shown in Table 4.

Table 4: Results (ROUGE-1/ROUGE-2/ROUGE-L F scores) of PEGASUS$_{\text{LARGE}}$ (mixed, stochastic) on downstream datasets. ‡ We updated the BIGPATENT dataset to preserve casing, some format cleanings are also changed.

XSum	CNN/DailyMail	NEWSROOM
47.60/24.83/39.64	44.16/21.56/41.30	45.98/34.20/42.18
Multi-News	Gigaword	WikiHow
47.65/18.75/24.95	39.65/20.47/36.76	46.39/22.12/38.41
Reddit TIFU	BIGPATENT	arXiv
27.99/9.81/22.94	52.29/33.08/41.66 ‡	44.21/16.95/25.67
PubMed	AESLC	BillSum
45.97/20.15/28.25	37.68/21.25/36.51	59.67/41.58/47.59

7 Conclusion

In this work, we proposed PEGASUS, a sequence-to-sequence model with gap-sentences generation as a pre-training objective tailored for abstractive text summarization. We studied several gap-sentence selection methods and identified principle sentence selection as the optimal strategy. We demonstrated the effects of the pre-training corpora, gap-sentences ratios, vocabulary sizes and scaled up the best configuration to achieve state-of-the-art results on all 12 diverse downstream datasets considered. We also showed that our model was able to adapt to unseen summarization datasets very quickly, achieving strong results in as little as 1000 examples. We finally showed our model summaries achieved human performance on multiple datasets using human evaluation.

8 Code and Model Checkpoints Release

The training code and instructions for using model checkpoints can be found at

`https://github.com/google-research/pegasus`

Acknowledgments

We thank Anastassia Kornilova, Eva Sharma, Shashi Narayan, Adam Roberts, Etienne Pot, and the Google News team for assistance with datasets, and Carey Radebaugh, David Grangier, Doug Eck, and Samy Bengio for reviewing the manuscript.

References

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. 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)*, pp. 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2097. URL https://www.aclweb.org/anthology/N18-2097.

Dai, A. M. and Le, Q. V. Semi-supervised sequence learning. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 3079–3087. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5949-semi-supervised-sequence-learning.pdf.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. Unified language model pre-training for natural language understanding and generation. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

Fabbri, A., Li, I., She, T., Li, S., and Radev, D. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1074–1084, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1102. URL https://www.aclweb.org/anthology/P19-1102.

Goodman, S., Lan, Z., and Soricut, R. Multi-stage pretraining for abstractive summarization, 2019.

Graff, D., Kong, J., Chen, K., and Maeda, K. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4 (1):34, 2003.

Grusky, M., Naaman, M., and Artzi, Y. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018. doi: 10.18653/v1/n18-1065. URL http://dx.doi.org/10.18653/v1/n18-1065.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pp. 1693–1701, 2015.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. SpanBERT: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*, 2019.

Khandelwal, U., Clark, K., Jurafsky, D., and Kaiser, L. Sample efficient text summarization using a single pre-trained transformer. *arXiv preprint arXiv:1905.08836*, 2019.

Kim, B., Kim, H., and Kim, G. Abstractive summarization of Reddit posts with multi-level memory networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2519–2531, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1260. URL https://www.aclweb.org/anthology/N19-1260.

Klimt, B. and Yang, Y. The enron corpus: A new dataset for email classification research. In *Proceedings of the 15th European Conference on Machine Learning*, ECML'04, pp. 217–226, Berlin, Heidelberg, 2004. Springer-Verlag. ISBN 3-540-23105-6, 978-3-540-23105-9. doi: 10.1007/978-3-540-30115-8_22. URL https://doi.org/10.1007/978-3-540-30115-8_22.

Kornilova, A. and Eidelman, V. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 48–56, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5406. URL https://www.aclweb.org/anthology/D19-5406.

Koupaee, M. and Wang, W. Y. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*, 2018.

Kryscinski, W., Keskar, N. S., McCann, B., Xiong, C., and Socher, R. Neural text summarization: A critical evaluation. 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)*, pp. 540–551, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1051. URL https://www.aclweb.org/anthology/D19-1051.

Kudo, T. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*, 2018.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hyg0vbWC-.

Nallapati, R., Zhou, B., dos Santos, C., Gulçehre, Ç., and Xiang, B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 280–290, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1028. URL https://www.aclweb.org/anthology/K16-1028.

Nallapati, R., Zhai, F., and Zhou, B. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 3075–3081. AAAI Press, 2017. URL http://dl.acm.org/citation.cfm?id=3298483.3298681.

Narayan, S., Cohen, S. B., and Lapata, M. 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*, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://www.aclweb.org/anthology/D18-1206.

Paulus, R., Xiong, C., and Socher, R. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018a.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2018b. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016. doi: 10.18653/v1/d16-1264. URL http://dx.doi.org/10.18653/v1/D16-1264.

Ramachandran, P., Liu, P., and Le, Q. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 383–391, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1039. URL https://www.aclweb.org/anthology/D17-1039.

Rothe, S., Narayan, S., and Severyn, A. Leveraging pre-trained checkpoints for sequence generation tasks. *arXiv preprint arXiv:1907.12461*, 2019.

Rush, A. M., Chopra, S., and Weston, J. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1044. URL https://www.aclweb.org/anthology/D15-1044.

See, A., Liu, P. J., and Manning, C. D. Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368, 2017. URL http://arxiv.org/abs/1704.04368.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://www.aclweb.org/anthology/P16-1162.

Sharma, E., Li, C., and Wang, L. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2204–2213, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1212. URL https://www.aclweb.org/anthology/P19-1212.

Shazeer, N. and Stern, M. Adafactor: Adaptive learning rates with sublinear memory cost. *arXiv preprint arXiv:1804.04235*, 2018.

Shi, T., Wang, P., and Reddy, C. K. LeafNATS: An open-source toolkit and live demo system for neural abstractive text summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 66–71, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4012. URL https://www.aclweb.org/anthology/N19-4012.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pp. 5926–5936, 2019.

Subramanian, S., Li, R., Pilault, J., and Pal, C. On extractive and abstractive neural document summarization with transformer language models. *arXiv preprint arXiv:1909.03186*, 2019.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 3104–3112, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969173.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Völske, M., Potthast, M., Syed, S., and Stein, B. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 59–63, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4508. URL https://www.aclweb.org/anthology/W17-4508.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018. doi: 10.18653/v1/w18-5446. URL http://dx.doi.org/10.18653/v1/w18-5446.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, pp. 5754–5764, 2019. URL http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf.

Zhang, R. and Tetreault, J. This email could save your life: Introducing the task of email subject line generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 446–456, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1043. URL https://www.aclweb.org/anthology/P19-1043.

Zhong, M., Liu, P., Wang, D., Qiu, X., and Huang, X. Searching for effective neural extractive summarization: What works and whats next. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. doi: 10.18653/v1/p19-1100. URL http://dx.doi.org/10.18653/v1/p19-1100.

A Datasets Statistics

Following Grusky et al., we calculate extractive fragment coverage/density for all downstream datasets. They were defined as

$$coverage = \frac{1}{S} \sum_{f \in F(A,S)} |f|$$

$$density = \frac{1}{S} \sum_{f \in F(A,S)} |f|^2$$

where A is article, S is summary, and $f \in F(A, S)$ are extractive fragments. High density indicates more extractive datasets and low coverage suggests more novel words in the summary.

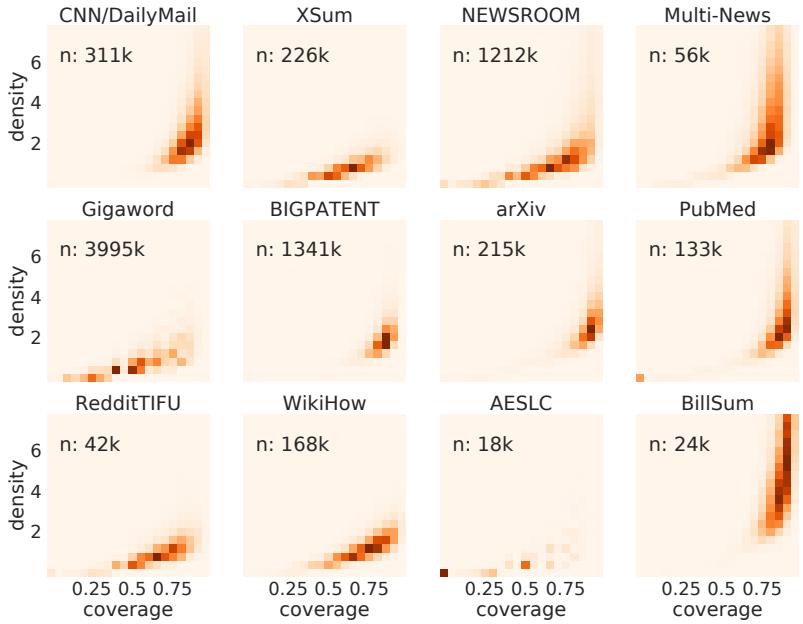


Figure A.1: A comparison of extractive fragment coverage and density of downstream datasets. The darker blocks indicate higher percentages and the n is the number of examples in the dataset.

B Pre-training Steps

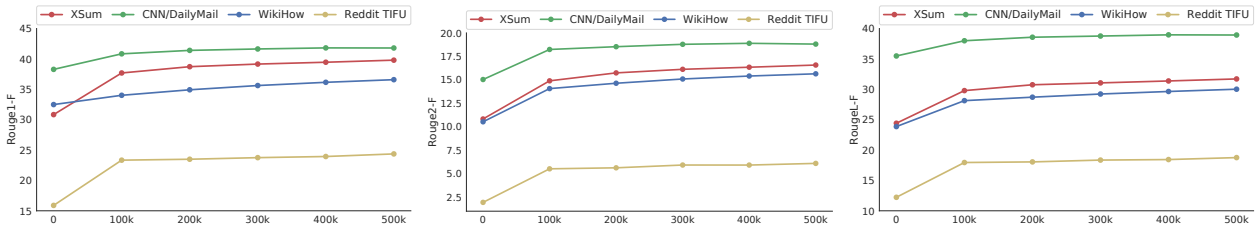


Figure B.1: Performance increase on downstream datasets as $\text{PEGASUS}_{\text{BASE}}$ trains for more steps on C4.

C PEGASUS Hyper Parameters

Table C.1: Hyperparamters of the pre-training and fine-tuning stages reported in section 6. The hyperparameters of fine-tuning $\text{PEGASUS}_{\text{LARGE}}$ were decided by grid search while others were decided by empirically default commonly used values. Max input/target tokens correspond to L_{input} and L_{target} in Section 6.

Pre-training (default unless otherwise specified in section 6)								
Model	Learning rate	Label smoothing	Num of steps	Batch size	Objective	Corpus	Max input tokens	Max target tokens
$\text{PEGASUS}_{\text{BASE}}$	0.1	0.0	500k	256	Ind-Orig	c4	512	256
$\text{PEGASUS}_{\text{LARGE}}$	0.1	0.0	500k	8192	Ind-Orig	c4 or HugeNews	512	256
Fine-tuning of $\text{PEGASUS}_{\text{BASE}}$ in Figure 3, 4, 5, B.1 and Table 1								
Dataset	Learning rate	Label smoothing	Num of steps	Batch size	Beam size	Beam alpha	Max input tokens	Max target tokens
XSum	5e-4	0.1	50k	256	1	-	512	64
CNN/DailyMail	5e-4	0.1	50k	256	1	-	512	128
NEWSROOM	5e-4	0.1	50k	256	1	-	512	128
Multi-News	5e-4	0.1	50k	256	1	-	512	256
WikiHow	5e-4	0.1	50k	256	1	-	512	256
Reddit TIFU	5e-4	0.1	50k	256	1	-	512	128
BIGPATENT	0.01	0.1	300k	256	1	-	512	256
arXiv	5e-4	0.1	50k	256	1	-	512	256
PubMed	5e-4	0.1	50k	256	1	-	512	256
Gigaword	5e-4	0.1	50k	256	1	-	128	32
AESLC	5e-4	0.1	50k	256	1	-	512	32
BillSum	5e-4	0.1	50k	256	1	-	512	256
$\text{Transformer}_{\text{BASE}}$ in Table 1								
Dataset	Learning rate	Label smoothing	Num of steps	Batch size	Beam size	Beam alpha	Max input tokens	Max target tokens
BIGPATENT	0.01	0.1	300k	256	1	-	512	256
AESLC	5e-4	0.1	300k	256	1	-	512	32
Others	5e-3	0.1	300k	256	1	-	Same as $\text{PEGASUS}_{\text{BASE}}$	
Fine-tuning of $\text{PEGASUS}_{\text{LARGE}}$ in Table 1 and 2								
Dataset	Learning rate	Label smoothing	Num of steps	Batch size	Beam size	Beam alpha	Max input tokens	Max target tokens
XSum(C4)	1e-4	0.1	130k	256	8	0.8	512	64
XSum(HugeNews)	1e-4	0.1	80k	256	8	0.8	512	64
CNN/DailyMail(C4)	5e-5	0.1	220k	256	8	0.8	1024	128
CNN/DailyMail(HugeNews)	5e-5	0.1	170k	256	8	0.9	1024	128
NEWSROOM	4e-4	0.1	104k	256	8	0.8	512	128
Multi-News	5e-5	0.1	80k	256	8	0.9	1024	256
WikiHow	8e-4	0.1	50k	256	8	0.6	512	256
Reddit TIFU	1e-4	0.1	12k	256	8	0.6	512	128
BIGPATENT	5e-3	0.1	300k	256	8	0.7	1024	256
arXiv	8e-4	0.1	74k	256	8	0.8	1024	256
PubMed	2e-4	0.1	100k	256	8	0.8	1024	256
Gigaword	8e-4	0.1	90k	256	8	0.6	128	32
AESLC	2e-4	0.1	16k	256	8	0.6	512	32
BillSum	2e-4	0.1	100k	256	8	0.8	1024	256
Fine-tuning of $\text{PEGASUS}_{\text{LARGE}}$ in Figure 6								
Dataset	Learning rate	Label smoothing	Num of steps	Batch size	Beam size	Beam alpha	Max input tokens	Max target tokens
all	5e-4	0.1	2k	256	1	-	Same as $\text{PEGASUS}_{\text{BASE}}$	

D Experiment Figures' Numbers

Table D.1: The raw ROUGE1-F1, ROUGE2-F1 and ROUGEL-F1 scores reported in corresponding figures.

ROUGE scores reported in Figure 3				
	XSum	CNN/DailyMail	WikiHow	Reddit TIFU
	R1/R2/RL	R1/R2/RL	R1/R2/RL	R1/R2/RL
Pre-trained on c4	39.79/16.58/31.70	41.79/18.81/38.93	36.58/15.64/30.01	24.36/6.09/18.75
Pre-trained on HugeNews	41.63/18.47/33.48	42.34/19.22/39.49	34.93/14.67/28.63	24.11/5.99/18.57
ROUGE scores reported in Figure 4a				
	XSum	CNN/DailyMail	WikiHow	Reddit TIFU
	R1/R2/RL	R1/R2/RL	R1/R2/RL	R1/R2/RL
Random	39.28/16.23/31.21	41.80/18.91/38.88	36.27/15.47/29.67	24.04/6.01/18.47
Lead	39.22/16.12/31.09	41.70/18.78/38.85	35.30/14.79/28.85	23.48/5.78/18.00
Ind-Orig	39.79/16.58/31.70	41.79/18.81/38.93	36.58/15.64/30.01	24.36/6.09/18.75
Ind-Uniq	39.50/16.41/31.41	41.79/18.83/38.94	36.26/15.47/29.69	24.10/5.98/18.41
Seq-Orig	39.22/16.27/31.11	41.88/18.89/39.02	36.39/15.57/29.74	24.09/6.15/18.55
Seq-Uniq	39.50/16.39/31.40	41.98/19.03/39.11	36.69/15.61/29.95	24.25/6.17/18.67
MLM solely	37.22/14.48/29.62	39.33/17.34/36.65	32.20/13.19/27.05	21.00/3.96/16.27
MLM & Ind-Orig	39.08/16.21/31.20	41.48/18.70/38.63	35.99/15.29/29.57	24.19/6.16/18.70
ROUGE scores reported in Figure 4b				
	XSum	CNN/DailyMail	WikiHow	Reddit TIFU
	R1/R2/RL	R1/R2/RL	R1/R2/RL	R1/R2/RL
15%	39.47/16.32/31.30	41.88/18.98/38.97	35.63/15.08/29.23	24.06/5.91/18.52
30%	39.61/16.51/31.48	41.83/18.82/38.96	36.26/15.47/29.69	24.05/6.05/18.55
45%	39.43/16.42/31.36	41.57/18.67/38.69	36.39/15.46/29.85	23.47/5.61/18.01
50%	39.19/16.20/31.16	41.49/18.60/38.64	36.15/15.36/29.56	23.92/5.83/18.33
60%	39.06/16.08/31.08	41.27/18.40/38.42	36.04/15.34/29.47	23.14/5.50/17.74
75%	36.94/14.21/29.14	40.17/17.52/37.37	34.32/13.72/27.96	21.72/4.32/16.45
ROUGE scores reported in Figure 5				
	XSum	CNN/DailyMail	WikiHow	Reddit TIFU
	R1/R2/RL	R1/R2/RL	R1/R2/RL	R1/R2/RL
BPE 32k	39.23/16.17/31.13	41.86/18.97/38.97	35.22/14.88/28.87	24.04/6.04/18.57
Unigram 32k	38.94/15.99/30.97	41.75/19.08/38.91	36.94/15.68/30.28	24.17/6.07/18.54
Unigram 64k	39.17/16.33/31.24	41.89/19.19/39.03	37.58/16.02/30.71	24.47/6.32/18.90
Unigram 96k	39.33/16.40/31.24	42.22/19.31/39.34	37.38/15.94/30.63	24.10/6.22/18.73
Unigram 128k	39.26/16.27/31.14	41.76/19.08/38.89	37.66/16.04/30.83	23.74/5.95/18.33
Unigram 256k	38.55/15.92/30.62	41.98/19.11/39.08	36.94/15.49/30.08	23.63/5.95/18.33
ROUGE scores reported in Figure B.1				
	XSum	CNN/DailyMail	WikiHow	Reddit TIFU
	R1/R2/RL	R1/R2/RL	R1/R2/RL	R1/R2/RL
No pretraining	30.83/10.83/24.41	38.27/15.03/35.48	32.48/10.53/23.86	15.89/1.94/12.22
100k-step	37.68/14.89/29.78	40.83/18.24/37.99	34.01/14.07/28.13	23.33/5.52/17.95
200k-step	38.72/15.74/30.74	41.40/18.53/38.57	34.91/14.64/28.70	23.48/5.62/18.05
300k-step	39.15/16.12/31.05	41.63/18.79/38.76	35.61/15.09/29.22	23.75/5.92/18.35
400k-step	39.45/16.34/31.37	41.81/18.89/38.95	36.14/15.41/29.64	23.93/5.92/18.43
500k-step	39.79/16.58/31.70	41.79/18.81/38.93	36.58/15.64/30.01	24.36/6.09/18.75

E Low Resource Numbers

Table E.1: The ROUGE1-F1, ROUGE2-F1 and ROUGEL-F1 scores of low resource summarization reported in Figure 6 along with previous SOTA in Table 1. With 100 examples, $\text{PEGASUS}_{\text{LARGE}}$ beats previous SOTA on ROUGE2-F1 metrics on BIGPATENT, Reddit TIFU, and BillSum dataset. With 1000 examples, $\text{PEGASUS}_{\text{LARGE}}$ beats previous SOTA metrics on Multi-News, WikiHow, Reddit TIFU, BigPatent, AESLC and BillSum.

Dataset	0 examples $R_1/R_2/R_L$	10 examples $R_1/R_2/R_L$	100 examples $R_1/R_2/R_L$	1k examples $R_1/R_2/R_L$	10k examples $R_1/R_2/R_L$	previous SOTA $R_1/R_2/R_L$
XSum	19.27/3.00/12.72	19.39/3.45/14.02	39.07/16.44/31.27	41.55/18.23/33.29	44.71/21.20/36.31	45.14/22.27/37.25
CNN/DailyMail	32.90/13.28/29.38	37.25/15.84/33.49	40.28/18.21/37.03	41.72/19.35/38.31	42.54/20.04/39.32	44.16/21.28/40.90
NEWSROOM	22.06/11.86/17.76	29.24/17.78/24.98	33.63/21.81/29.64	37.26/25.34/33.12	39.54/27.25/35.45	39.91/28.38/36.87
Multi-News	36.54/10.52/18.67	39.79/12.56/20.06	41.04/13.88/21.52	44.00/15.45/22.67	44.70/16.57/23.43	43.47/14.89/17.41
Gigaword	23.39/7.59/20.20	25.32/8.88/22.55	29.71/12.44/27.30	32.95/13.90/30.10	35.13/16.36/32.61	38.73/19.71/35.96
WikiHow	22.59/6.10/14.44	23.95/6.54/15.33	25.24/7.52/17.79	34.35/12.17/25.84	37.22/14.41/29.15	28.53/9.23/26.54
Reddit TIFU	14.66/3.06/10.17	15.36/2.91/10.76	16.64/4.09/12.92	23.34/6.85/18.46	25.47/8.18/20.33	19.0/3.7/15.1
BIGPATENT	25.61/6.56/17.42	28.87/8.30/19.71	33.52/10.82/22.87	36.85/12.58/24.54	34.81/12.39/24.13	37.52/10.63/22.79
arXiv	28.05/6.63/17.72	31.38/8.16/17.97	33.06/9.66/20.11	39.46/12.38/22.20	40.24/14.04/23.11	41.59/14.26/23.55
PubMed	28.17/7.57/17.85	33.31/10.58/20.05	34.05/12.75/21.12	40.15/15.56/24.05	41.75/16.74/24.80	40.59/15.59/23.59
AESLC	10.35/3.86/9.29	11.97/4.91/10.84	16.05/7.20/15.32	28.58/15.45/28.14	36.47/20.85/35.53	23.67/10.29/23.44
BillSum	41.02/17.44/25.24	40.48/18.49/27.27	44.78/26.40/34.40	46.47/30.58/37.21	50.81/34.49/40.96	40.80/23.83/33.73

F Human Evaluation Details

In all human evaluation experiments we used the same task template shown in Figure F.1, where workers were asked to rate 4 summaries for a document on a scale of 1 (poor summary) to 5 (great summary). The order in which the summaries are presented for each task was random per example. Each task was independently done by 3 different workers and we retained the median score across workers for each summary. We paid 1 USD per task and used the following critieria for workers to ensure high-quality:

- Location: US
- Minimum approval rate: 95%
- Minimum HIITs: 1000

With this criteria we observed high reproducibility in the conclusions of the huamn evaluation. Multiple runs of the same experiment with different workers meeting this criteria yielded very similar results. The HITT template is provided at `https://github.com/google-research/pegasus`.

In experiment 1, the four summaries corresponded to 3 models ($\text{PEGASUS}_{\text{LARGE}}$ pre-trained on HugeNews, C4, and $\text{Transformer}_{\text{BASE}}$) that were fine-tuned using all the supervised examples along with the reference (human) summary. We sampled 100 examples from each dataset (XSum, CNN/DailyMail, Reddit TIFU).

In experiment 2, we evaluated 4 models ($\text{PEGASUS}_{\text{LARGE}}$ pre-trained on HugeNews fine-tuned using different amounts of supervision, 10, 100, 1000, and all examples) alongside the human summary. To do this with the same template, for each example we randomly selected 4 out of the 5 summaries. This resulted in fewer ratings per model, but did not increase the work (and cost) of the task.

We used a paired t-test to determine statistical significance when comparing the ratings of two sets of summaries.

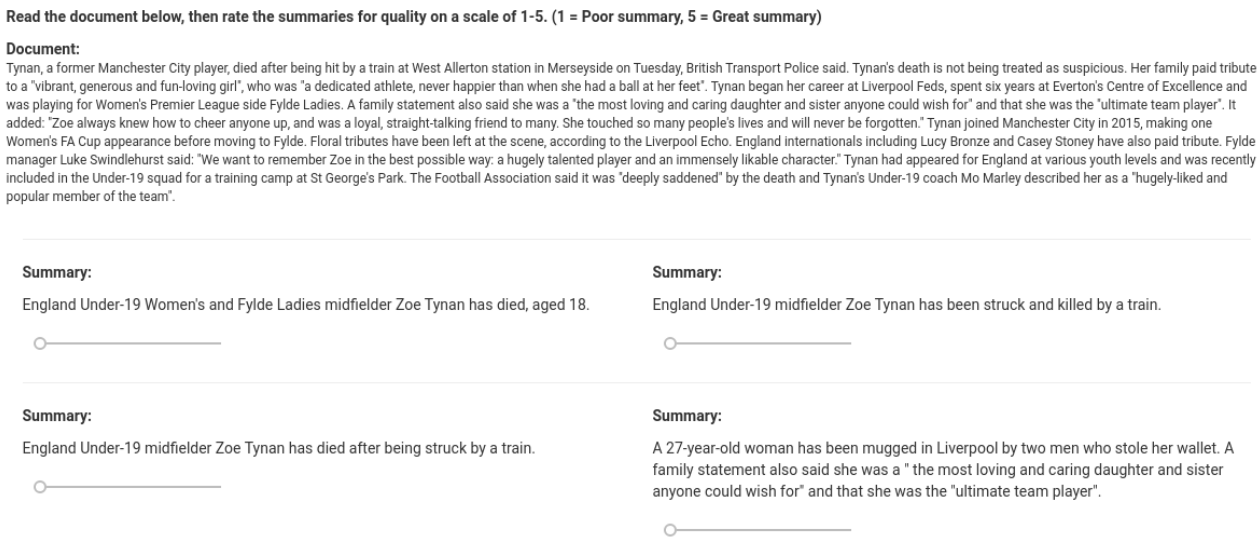


Figure F.1: A screenshot of the Amazon MTurk HIIT.

G Example of summary with relatively low ROUGE2-F but qualitatively good.

This figure shows an example model summary from the CNN/DailyMail dataset exhibiting high fluency, coherence, although highly abstractive, and only ROUGE2-F of 16. The model understood that the football team "Chelsea" could be paraphrased as "Jose Mourinho's side" and "The Blues" and highlighted the same four matches to be played.

Document: chelsea will face paris saint-germain, the french team who knocked jose mourinhos side out of the champions league this season, in a pre-season friendly in july. the blues, who were sent crashing out on away goals at the last-16 stage following a 2-2 draw at stamford bridge, will play psg in north carolina on july 25. it is one of three games mourinhos side will feature in across the pond as they gear up to defend a probable premier league title. john terry leads the celebrations as chelsea close in on the premier league title with a 0-0 draw at arsenal . eden hazard, the pfa player of the year, will line-up for chelsea when they travel to the usa in the summer . new york red bulls - july 22 - new jersey . paris saint-germain - july 25 - charlotte, north carolina . barcelona - july 28 - washington d.c. fiorentina - august 5 - stamford bridge . chelsea, 10 points ahead of arsenal with just four games to play, will also face the new york red bulls on july 22 and spanish giants barcelona six days later in washington. chelsea fans will then get to see their side before the premier league campaign kicks-off with a friendly against fiorentina at stamford bridge on august 5. all four matches mark chelseas participation in this summers pre-season international champions cup with manchester united, who mourinhos side will not face, la galaxy, porto and san jose earthquakes also involved. im pleased we are able to announce our fixtures for what promises to be an exciting summer,' said chelsea chairman bruce buck. as promised, we face some excellent opposition across several iconic venues in the united states and to top it off we are delighted to be hosting fiorentina at stamford ... ...

Ground-truth: chelsea to play three matches inside six days in the united states . they will face new york red bulls, paris saint-germain and barcelona . fiorentina will then travel to stamford bridge for friendly on august 5 . four matches will make up chelsea's participation in champions cup . read: chelsea interested in 43m antoine griezmann .

Model: jose mourinho's side will play psg in north carolina on july 25 . chelsea will also face the new york red bulls and barcelona . the blues will play fiorentina at stamford bridge on august 5 .

Figure G.1: A CNN/DailyMail $\text{PEGASUS}_{\text{LARGE}}$ model summary with relatively low ROUGE2-F of 16, but qualitatively quite good, and factually accurate.

H Abstractiveness of Summaries

We compared the abstractiveness of model generated summaries with the human-written ones for all downstream datasets. We measured abstractiveness of summaries using average values of extractive coverage and extractive density (Grusky et al., 2018) on each dataset. More abstractive summaries have smaller extractive coverage (more novel words) and smaller extractive density (smaller spans copied from inputs). Figure H.1 shows that the summaries generated by models were all less abstractive than the human-written counterparts. However, the models that were finetuned on more abstractive datasets, such as XSum and Reddit TIFU, could generate more abstractive summaries than human-written ones on other datasets.

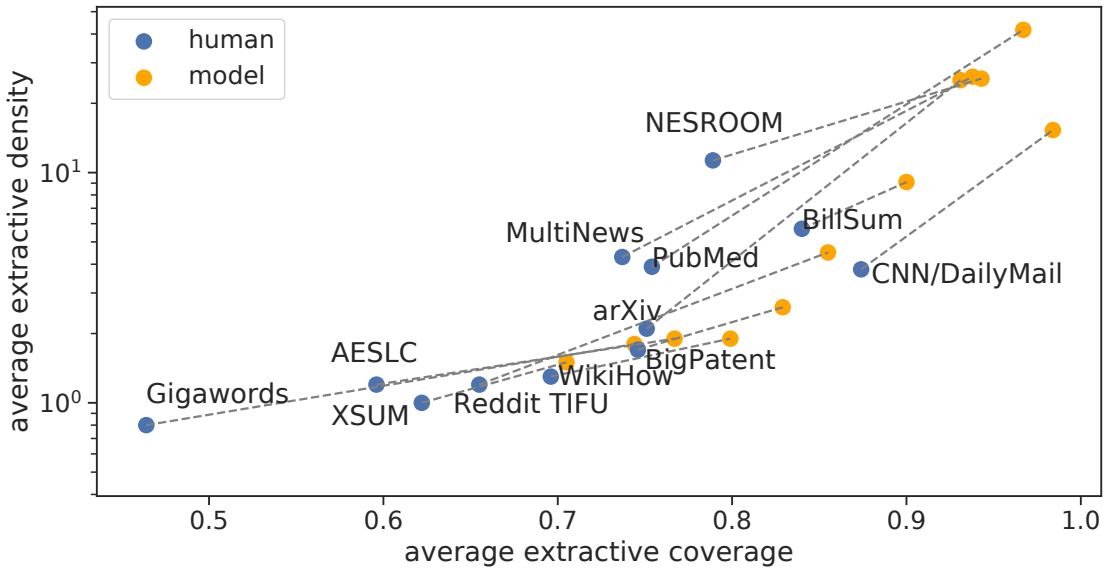


Figure H.1: Comparison of abstractiveness of human written and model generated summaries.

I Example Model Outputs

Model outputs were selected (and LaTeX tables generated) automatically by a program in the following way: (1) pick first 300 examples of triplets (document, gold summary, model output) from the dataset test split; (2) rank the examples by ROUGE1-F1/ROUGE2-F1/ROUGEL-F1 metrics in descending order; (3) divide the examples into 2-10 buckets depending on the documents lengths; (4) randomly pick one example from each bucket.

We filtered out examples that contain bad words from the link [3]. Input documents were truncated at 300 words for visualization. Each page shows examples from one dataset sampled by one ROUGE metric.

[3] https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words/blob/master/en