Repurposing Decoder-Transformer Language Models for Abstractive Summarization

Luke de Oliveira
Twilio AI
ldeoliveira@twilio.com

Alfredo Láinez Rodrigo
Twilio AI
alainez@twilio.com

Abstract

Neural network models have shown excellent fluency and performance when applied to abstractive summarization. Many approaches to neural abstractive summarization involve the introduction of significant inductive bias, such as pointer-generator architectures, coverage, and partially extractive procedures, designed to mimic human summarization. We show that it is possible to attain competitive performance by instead directly viewing summarization as language modeling. We introduce a simple procedure built upon pre-trained decoder-transformers to obtain competitive ROUGE scores using a language modeling loss alone, with no beam-search or other decoding-time optimization, and instead rely on efficient nucleus sampling and greedy decoding.

1 Introduction

Neural network approaches to abstractive summarization generally encode the source document into some hidden state or representation, then decode this representation into a summarized, abstracted version of the source document [17]. These approaches usually rely on a sequence-to-sequence [20] style architecture, and tend to produce fluent, well formed natural language summaries when coupled with beam search or other decoding techniques.

A weakness of traditional sequence-to-sequence learning when applied to summarization is the lack of a direct copy mechanism, leading to missing or misrepresented details in decoded summaries [2, 17]. Though attention helps ameliorate this issue by directly learning to focus on specific words or phrases in a source document [2], many have allowed for an explicit copy mechanism inspired by Pointer Networks [22], by optimizing a differentiable decision whether to generate new text or directly copy from the source [5, 18].

Peters et al. [15], Devlin et al. [3], Radford et al. [16], and many others have shown the benefits of large-scale pretraining on large, unlabeled corpora on a variety of downstream tasks in transfer learning settings. In particular, it has been shown that large-scale, attention-only language modeling via decoder-only transformers [11] as an unsupervised pretraining task admits the ability to perform zero-shot learning on meaningful tasks involving natural language generation [16].

Motivated by this, we propose a simple method that exhibits competitive performance on abstractive summarization without using sequence-to-sequence architectures or other standard tools in the neural abstractive summarization toolbox, and instead using a decoder-only transformer language model with transfer learning. This further illustrates the utility of finetuning language models trained on open domain text.

2 Model

Transformer Preliminaries Our model builds on previous work utilizing decoder-only Transformers [11] for jointly learning language modeling and sequence transduction in aligned domains,

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

which limits attention to tokens $0, 1, \ldots, n-1$ for predicting token n. Formally, a decoder-only Transformer considers a sequence of one-hot token vectors $T = [t_0, t_1, \ldots, t_{n-1}] \in \{0,1\}^{V \times n}$, with each $t_i \in \{0,1\}^V$ where V is the size of the vocabulary. Given an embedding matrix $W_E \in \mathbb{R}^{d \times V}$ and a positional encoding matrix $W_P \in \mathbb{R}^{d \times (n-1)}$, the model computes

$$H_0 = W_E T + W_P,\ H_\ell = \mathsf{TRF}(H_{\ell-1}) \in \mathbb{R}^{d \times (n-1)}, \forall \ell \in [1, \ldots, L] \tag{1}$$

where TRF is the transformer block with self-attention, first introduced in Vaswani et al. [21]. We utilize the modifications provided in Radford et al. [16], such as moving Layer Normalization [9] to the beginning of each transformer block.

Decoder-only Sequence Transduction for Summarization Formally, consider a set of paired documents $\mathcal{C} = \{(x, y)\}, |\mathcal{C}| = N$. For a source-summary pair $(x, y) \in \mathcal{C}$, the source document $x = [x_0, \ldots, x_m]$ and reference summary $y = [y_0, \ldots, y_k]$ are sequences of one-hot token vectors.

To learn this mapping using a language model, we combine x and y using special learnable vectors corresponding to control tokens. In addition, we augment Eq. 1 to include a *segment-specific* (i.e., source or summary) embedding [3]. Finally, we reset the positional encoding for the summary. Our model is fed three sequences (see Eq. 2): a concatenation of the source document and the summary (S), positional encodings that reset for the summary component (P), and segment-specific encodings for the source and the summary (Q). We represent the start of the source document with α, the beginning of the summary with β, and the end of sequence with δ. Additionally, we encode the source segment with σ and the summary segment with τ.

$$\begin{aligned} S &= [\alpha, x_0, \ldots, x_m, \beta, y_0, \ldots, y_k, \delta] \\ P &= [0, 1, \ldots, m, m+1, 0, 1, \ldots, k, k+1, 0] \\ Q &= [\sigma, \sigma, \ldots, \sigma, \sigma, \tau, \ldots, \tau, \tau] \end{aligned} \tag{2}$$

Thus, our model changes Eq. 1 by adding the position encoding modification from Eq. 2 and an additional trainable weight W_Q representing the segment encoding Q. The model is trained via maximum likelihood, where we take into account the full likelihood of the source-summary pair.

Input Representation Given recent trends moving away from purely word- or character-level representations, we utilize data-driven subword encoding via Byte Pair Encoding (BPE) [19], following the procedure outlined in Radford et al. [16]. For experiments in which we finetune the 117M parameter model from Radford et al. [16], we utilize their prebuilt vocabulary; in ablation studies, we utilize SentencePiece [8] to learn BPE merges.

3 Experimental Setup

Datasets We train and evaluate our models on the CNN/Daily Mail (CNN-DM) corpus [12] of news articles and summaries, utilizing the non-anonymized version [18]. We use the predefined training, validation, and test splits, and limit source articles to 400 tokens and summaries to 100 tokens at training time.

As an additional test, we train and evaluate the best model configuration from the ablation studies above on the Extreme Summarization (XSum) corpus [13], which contains single sentence summaries of BBC articles. As shown in Narayan et al. [13], the XSum corpus requires models to perform a much higher degree of semantic distillation, as indicated by low n-gram overlap, high n-gram novelty, and poorly performing LEAD-3 baselines.

Models & Inference We conduct experiments in two regimes for CNN-DM: first, we *finetune* the model outlined in Sec. 2 on top of the 117M parameter model release from Radford et al. [16], and second, we perform a full training from scratch in order to ablate the effect of transfer learning. We utilize a context size of 1024 with an embedding dimension of 768, 12 attention heads, and a batch size of 10. We train using the Adam [7] optimizer with a learning rate of 5×10^{-5} until the loss ceases to decrease on the validation set. For XSum, we use the highest-performing setup from CNN-DM experiments.

In lieu of beam search, we compare greedy decoding and nucleus sampling [6]. In both cases, we decode until we reach the stop-token δ (Eq. 2). In the case of nucleus sampling, we perform 5

Method	ROUGE-1	ROUGE-2	ROUGE-L
Pointer-Generator + Coverage [18]	39.53	17.28	36.38
ML + RL [14]	39.87	15.82	36.90
Bottom-Up [4]	41.22	18.68	38.34
DCA (best) [1]	41.69	19.47	37.92
GPT-2 `TL;DR` [16]	29.34	8.27	26.58
D-TRF (Finetuned + greedy, **ours**)	39.12	17.12	27.22
D-TRF (Finetuned + nucleus, **ours**)	40.70	18.03	29.62

Table 1: Comparison of our method with select existing methods on the CNN-DM dataset.

independent decodings[1] with $p = 0.3$, then pick the decoding that reports the lowest negative log likelihood score of the *completed summary*. We use $1/k^{0.6}$ as a likelihood normalization term to avoid a preference for shorter summaries, borrowing directly from Wu et al. [23].

Ablation	R-1	R-2	R-L
Best	40.70	18.03	29.62
(–) Finetuning	36.10	15.06	26.92
(–) Segment encoding	38.80	16.33	27.19

Table 2: Ablation of model components on CNN-DM (Decoded via nucleus sampling procedure).

Evaluation We evaluate all models using the ROUGE metric [10], in particular the F1 variants of ROUGE-1, ROUGE-2, and ROUGE-L which measure unigram overlap, bigram overlap, and longest common subsequence respectively.

4 Results

CNN-DM Our main results are displayed in Table 1, where we compare our method (in the bottom section of the table) to existing methods (in the upper portion) on the CNN-DM dataset, and show ablations in Table 2.

We note that our models (for ROUGE-1 and -2) are competitive even when using greedy decoding, and without any sequence-to-sequence style architectures or coverage terms, illustrating the power of this approach for abstractive summarization. We note that using a well trained language model [16] and then finetuning yields a significant performance jump (as shown via ablation in Table 2), motivating this method in practical contexts given the recent trends toward large-scale, self-supervised learning approaches [3, 16, 15].

XSum As a secondary evaluation of our approach, we train our best model on the XSum dataset [13] and report ROUGE scores in a direct comparison to the benchmarks reported. Results for these experiments are shown in Table 3. We achieve highly competitive performance relative to models reported in Narayan et al. [13] building on a finetuning approach without using many of the inductive biases traditionally present in summarization methods.

5 Conclusion

This work puts forward a simple approach to abstractive summarization by viewing sequence transduction as a language modeling problem. We show the effectiveness of using decoder-only transformers for this task, in particular, when coupled with recent advances in large-scale language modeling and transfer learning. We show that competitive performance on two benchmark datasets is possible without many of the standard tools in neural abstractive summarization, such as sequence-to-sequence modeling, coverage mechanisms, direct ROUGE optimization via reinforcement learning, or beam search, instead relying on a purely language modeling loss and simple decoding mechanisms such as nucleus sampling and greedy decoding. This approach yields highly fluent text, and illustrates the power of unsupervised representation learning-based transfer learning for downstream tasks.

[1]Nucleus sampling with $p = 0.3$ implies we only sample from the top 30% of of the probability distribution over tokens

Method	R-1	R-2	R-L
Seq2Seq Baseline	28.42	8.77	22.48
Conv-Seq2Seq	31.27	11.07	25.23
Topic-ConvSeq2Seq	31.89	11.54	25.75
D-TRF (Finetuned + nucleus)	34.19	12.17	27.06

Table 3: Comparison of with existing methods on XSum, reported in Narayan et al. [13].

References

[1] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. Deep communicating agents for abstractive summarization. *arXiv preprint arXiv:1803.10357*, 2018.

[2] Sumit Chopra, Michael Auli, Alexander M Rush, and SEAS Harvard. Abstractive sentence summarization with attentive recurrent neural networks. *Proceedings of NAACL-HLT16*, pages 93–98, 2016.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[4] Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[5] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*, 2016.

[6] Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration. *arXiv e-prints*, art. arXiv:1904.09751, Apr 2019.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[9] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *arXiv e-prints*, art. arXiv:1607.06450, Jul 2016.

[10] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, Barcelona, Spain, July 2004.

[11] Peter J. Liu, Mohammad Ahmad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. 2018.

[12] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[13] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *EMNLP 2018*.

[14] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

[15] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL 2018*.

[16] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[17] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[18] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of ACL 2017*.

[19] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[20] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[22] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700, 2015.

[23] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.