Learning to Prioritize: Precision-Driven Sentence Filtering for Long Text Summarization

Anonymous ACL submission

Abstract

Neural text summarization has shown great potential in recent years. However, current state-of-the-art summarization models are limited by their maximum input length, posing a challenge to summarize longer texts comprehensively. As part of a layered summarization architecture, we introduce PURETEXT, a simple yet effective precision-driven sentence filtering layer that learns to remove low-quality sentences in texts to improve existing summarization models. When evaluated on popular datasets like WikiHow and Reddit TIFU, we show up to 3 and 8 point ROUGE-1 absolute improvement on the full test set and the long article subset, respectively, for state-of-the-art summarization models such as BERTSUM and BART. Our approach provides downstream models with higher-quality sentences for summarization, improving overall model performance, especially on long text articles.

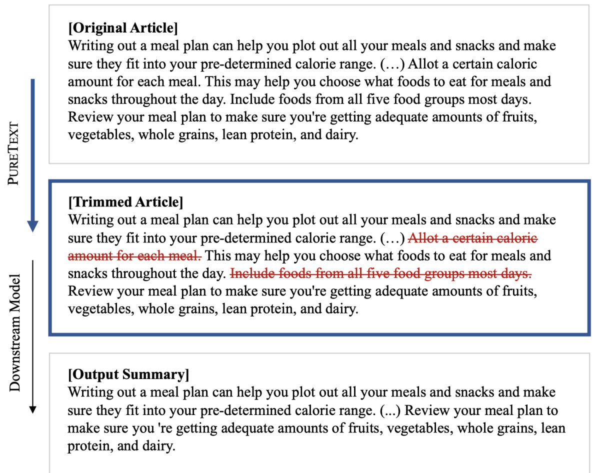

Figure 1: A WikiHow instructional article on "How to Lose Weight Without Exercising." Rather than feeding the article directly to a model for summarization, we first filter high-quality sentences using a weakly-supervised layer that we call PURETEXT.

1 Introduction

Neural summarization models have evolved quickly over time, proving successful in tackling increasingly complex problems relating to natural language (Zhong et al., 2020; Zhou et al., 2018; Zhang et al., 2019; Xu et al., 2020). One key problem that has plagued state-of-the-art summarization models is their maximum input length (Liu, 2019; Lewis et al., 2020). Although recent work has made progress towards addressing this issue for Transformer-based models (Beltagy et al., 2020; Zhou et al., 2021; Choromanski et al., 2021), not as much attention has been paid specifically towards long text summarization.

Summarization models such as BERTSUM (Liu, 2019) and BART (Lewis et al., 2020) either truncate or cannot handle articles longer than the maximum input length. Truncation may leave out critical parts of the text, leading to an incomplete summary.

For datasets where LEAD-3 forms a decent baseline (Nallapati et al., 2016; Narayan et al., 2018), truncating an article's ending may not greatly affect summarization. While this may be true for news summarization datasets in which story highlights tend to appear at the start (Hermann et al., 2015; Nallapati et al., 2016; Narayan et al., 2018), other datasets such as WikiHow (Koupaee and Wang, 2018) and Reddit TIFU (Kim et al., 2019) typically do not follow the same journalistic structure.

WikiHow instructional texts contain key steps evenly dispersed throughout the article, and Reddit stories tend to follow a narrative arc where the climax is toward the end of the passage.

One simple solution to truncation is to omit the middle section of an article instead (Sun et al., 2019). However, this method, along with similar approaches, is a heuristic that can potentially be improved upon with a more versatile model.

While existing work shows promising results for long text summarization (Beltagy et al., 2020; Xu et al., 2018), they require extensive computational resources to run. Instead, we propose a lightweight weakly-supervised layer to prepend to state-of-the-art summarization models to improve their performance on long text summarization. Our process is a two-step summarization scheme in which we first apply a filtering layer which serves as a screen for high quality sentences, and then summarize using a state-of-the-art summarization model that produces the final refined summary. Although other multi-step long text summarization processes have been attempted in the past, they often have specific applications like in low resource settings (Bajaj et al., 2021) or documents with an identifiable discourse structure (Gidiotis and Tsoumakas, 2020). Other methods are non-generalizable to already existing summarization systems (Wang et al., 2017). Our layered summarization architecture allows for versatility, as the filtering layer can be used to augment many existing downstream summarization models.

Our filtering layer takes inspiration from dense sentence retrieval, (Zhong et al., 2020; Zhang et al., 2019) prioritizing important sentences for summarization. Critically, we take a weakly-supervised learning approach in which we train a BERT-based model to rank the importance of sentences based on their individual ROUGE scores when compared with the gold summary. We then filter up to 80% of an article's sentences before feeding it to a downstream summarization model. Figure 1 provides an example of our full pipeline on a single article.

We experiment on the WikiHow and Reddit TIFU datasets and observe that our model removes sentences irrelevant for summarization, improving on previous state-of-the-art results.

To summarize our contributions:

- We propose a model-agnostic weakly supervised learning objective using text similarity.
- We explore a layered-architecture approach in text summarization and introduce a versatile, lightweight filtering layer that we name PURE-TEXT for filtering out low-quality sentences.
- We test our approach on BERTSUM and BART and find up to 3 point ROUGE-1 improvement on the WikiHow and Reddit datasets.

2 Methodology

We fine-tune a BERT-based model [1] to classify sentences as either "important" or "unimportant" using a sentence's ROUGE-1 F_1 score to generate its label. We choose to classify at the sentence level because sentences are a natural subunit of an article with self-containing grammar. We assume that a sentence's ROUGE-1 F_1 score is strongly correlated with its degree of importance for summarization since ROUGE-1 F_1 is the final metric used for evaluation. Subsequently, we select the best subset of sentences that do not exceed the downstream model token limit and then feed the filtered article to a downstream model for summarization. We further experiment to see whether additional filtration beyond the downstream model input limit helps further improve summary quality.

3 Classification

To supervise the training of the classifier, we create silver labels consisting of either "important" or "unimportant" for each sentence. To determine the importance of each sentence in the article, we utilize ROUGE due to its lightweight text similarity measure. Specifically, for a given sentence, we first calculate its ROUGE-1 F_1 similarity score to the ground-truth summary. We then label a percentage of the sentences with the highest score as "important" and the rest as "unimportant". After varying the ratio of "important" to "unimportant" sentences in increments of 10%, we find that labeling sentences with a score above the median [2] as "important" and sentences with a score below the median as "unimportant" works best.

We tested ROUGE-1 precision and recall as alternative labelling metrics to F_1, but found that ROUGE-1 F_1 produced the best scoring summaries. Since extractive models can maximize recall by using the entire article as a summary, F_1 provides a balance by taking the harmonic mean of recall and precision. Thus, we take a precision-driven approach to maximize the final ROUGE-1 F_1 scores.

Once we generate the labels for each of the sentences in our training set, we train our BERT-based classifier and then use it to predict the importance of sentences in our test set.

[1] We use the BERT Sequence Classification model from Hugging Face for 5 epochs using early stopping, learning rate = $1 * 10^{-6}$, weight decay = 0.005, warmup steps = 0, and batch size = 32. Checkpoints are saved every 250 steps and we choose the model checkpoint with the lowest validation loss. For all other hyperparameters, we use the default provided by Hugging Face Trainer. The model is trained on 3 NVIDIA Titan X Pascal + 1 GeForce GTX Titan X GPUs for 10,000 steps each, elapsing 10 hours on average.

[2] We calculate the median score for each article to assign labels for each sentence within it. This way, we ensure that each article consists of an equal number of "important" and "unimportant" sentences.

$R_1/R_2/R_L$	WikiHow$_{full}$		Reddit TIFU$_{full}$	
Method	**BERTSUM**	**BART**	**BERTSUM**	**BART**
BASELINE	30.70/8.77/28.54	23.30/5.74/15.14	20.88/5.14/17.18	13.10/2.30/9.17
RANDOM$_{50}$	28.95/7.64/26.86	24.43/5.81/15.39	19.35/4.10/15.86	14.78/2.59/10.11
PURETEXT$_{default}$	**31.53**/**9.10**/**29.30**	23.47/5.81/15.22	20.98/5.25/17.30	13.18/2.33/9.21
PURETEXT$_{20}$	**31.53**/9.07/29.27	23.63/5.86/15.24	**21.03**/**5.32**/**17.33**	13.26/2.36/9.26
PURETEXT$_{80}$	29.52/7.82/27.19	**27.14**/**7.05**/**16.62**	19.32/4.42/15.60	**15.85**/**3.17**/**10.77**

Table 1: ROUGE F_1 scores produced by downstream summarization models on the full test sets when we apply our sentence filtering approach, labeling 50% of sentences as "important". We apply additional filtration, denoted by PURETEXT$_{default}$ (filtering to the maximum input limit) or PURETEXT$_x$ (filtering to $x\%$ below the maximum input limit, e.g. PURETEXT$_{20}$ would mean filtering to 410 tokens rather than 512 for BERTSUM). We compare to baselines without filtering as well as a 50% random filtering. The results we present are statistically significant with $\rho < 0.05$.

$R_1/R_2/R_L$	WikiHow$_{subset}$		Reddit TIFU$_{subset}$	
Method	**BERTSUM**	**BART**	**BERTSUM**	**BART**
BASELINE	30.12/8.07/28.23	22.46/4.35/14.62	20.52/3.91/16.52	11.26/1.13/8.27
RANDOM$_{50}$	30.25/8.01/28.26	23.44/4.73/14.70	20.26/3.68/16.43	13.06/1.52/9.12
PURETEXT$_{default}$	32.33/**8.95**/**30.25**	23.55/4.85/15.21	20.98/**5.25**/**17.30**	12.64/1.59/8.95
PURETEXT$_{20}$	**32.40**/8.85/30.25	24.39/5.10/15.23	**21.20**/4.65/17.23	14.12/1.99/9.70
PURETEXT$_{80}$	30.00/7.36/27.78	**31.03**/**7.11**/**17.24**	19.90/3.83/15.82	**17.39**/**2.96**/**11.36**

Table 2: ROUGE F_1 scores produced by downstream summarization models on the subset of long articles from the test sets. Other variables are consistent with those in Table 1.

3.1 Sentence Selection

After the classifier predicts each sentence as either "important" or "unimportant," the sentences of each article are ranked by their respective probabilities of being "important." Since the training objective of the model is to maximize the ROUGE-1 F_1 score, we define the reward R_i of a given sentence based on its probability of falling into the "important" class as assigned by the model.

Next, we frame the problem of finding the set of sentences that produce the highest cumulative reward, $\sum R_i$, without exceeding the given token limit L [3] of a downstream model in the context of the 0-1 Knapsack algorithm. Each sentence is weighted according to its number of tokens. Finally, we feed the best set of sentences, which we call the "trimmed article," to a downstream model for summarization.

3.2 Sentence Filtration

The 0-1 Knapsack algorithm finds the most important sentences up to the token limit. At the same time, we hypothesize that filtering additional low-quality sentences can benefit the downstream summarization model by providing a better signal. We grid-search from 0 to 80% additional filtering below the maximum input token limit L to determine the best percentage.

4 Resources

We choose to test our method on the WikiHow and Reddit TIFU datasets due to their non-journalistic structure. We also examine the results on the subset of long text articles within these datasets since that is where we aim to see the most improvement. Additionally, we select downstream models with the ability to analyze texts at a finer granularity than the sentence level so that the final outputted summary can be further refined beyond our best-selected sentences.

WikiHow (Koupaee and Wang, 2018) is an instructional text dataset. It contains 180K step-by-step tutorials with a summarizing sentence and a detailed paragraph elaboration for each instruction.

Reddit TIFU (Kim et al., 2019) is a summarization dataset. We use only the TIFU-long subset, which contains 40K posts from the TIFU subreddit. Each post contains a "TL;DR" as the summary.

[3] L is 512 for BERTSUM and 1024 for BART.

[Ground Truth Summary]
Count calories. Write yourself a meal plan. Eat a balanced diet. Snack healthy. Choose healthier cooking techniques. Drink adequate amounts of fluids. Ditch alcohol and sugary beverages.

[Output Summary w/ PURETEXT]
Writing out a meal plan can help you plot out all your meals and snacks and make sure they fit into your pre-determined calorie range. Figure out how many calories you can cut from your daily diet by first calculating the number of calories you should take in each day. Review your meal plan to make sure you're getting adequate amounts of fruits, vegetables, whole grains, lean protein, and dairy.

[Output Summary w/o PURETEXT]
Writing out a meal plan can help you plot out all your meals and snacks and make sure they fit into your pre-determined calorie range. Figure out how many calories you can cut from your daily diet by first calculating the number of calories you should take in each day. Weight loss programs usually require you to modify your total calorie intake.

Figure 2: An example of a summary generated with and without PURETEXT as compared with the Ground Truth Summary, using the same article from Figure 1. The summary produced without PURETEXT includes an irrelevant sentence, while the output summary with PURETEXT includes a relevant sentence that would have otherwise been truncated.

BERTSUM (Liu, 2019) is a fine-tuned BERT model for extractive summarization with the ability to perform trigram blocking.

BART (Lewis et al., 2020) is an autoencoder for pretraining sequence-to-sequence models using bidirectional and auto-regressive transformers. We use the standard, non-fine-tuned, version of BART to show that our sentence filtering approach does not require downstream models to be fine-tuned.

5 Results

We evaluate PURETEXT's performance on WikiHow and Reddit using BERTSUM and BART. Notably, we see strong relative improvements in downstream summary quality for BERTSUM and BART with PURETEXT. These results are compared with two baselines: summarization without PURETEXT (i.e. article is fed directly to the downstream summarization model) and summarization with random dropping. For the random baseline, each sentence has a 50% chance of being removed.

5.1 Full Dataset

We present the results from evaluating PURETEXT with multiple levels of additional filtration on the full WikiHow and Reddit datasets in Table 1. Note that we also experimented with the CNNDM and XSum news datasets and found statistically insignificant results. We find that BERTSUM and BART improve up to 0.83 and 3.84 points in absolute ROUGE-1 F_1, respectively, when compared to the baseline summaries.

Since out-of-the-box BART is not fine-tuned for a specific dataset, we must provide additional support to guide the model. To provide better signal, we apply additional filtering to further remove lower quality sentences. For fine-tuned BERTSUM, however, it learns to utilize context from lower quality sentences to improve the overall summary quality with less filtration.

5.2 Long Article Subset

Since PURETEXT aims to improve summarization on longer articles, we manually construct a subset of each dataset containing only articles that exceed the downstream model input limit. To explore these results, we consider both a qualitative and quantitative evaluation. Figure 2 shows qualitatively that PURETEXT enables downstream models to summarize with better context, as opposed to the default arbitrary truncation. Table 2 shows PURETEXT improves on the long article subset by a factor of 3 greater than the full dataset, with up to a 2.28 and 8.57 point improvement on BERTSUM and BART respectively. These improvements provide statistically significant evidence that PURETEXT improves long text summarization.

6 Conclusion

We introduce a novel, precision-driven sentence filtering layer called PURETEXT. We utilize a BERT-based model trained with weakly-supervised learning to distinguish high-quality sentences, which are then passed to a state-of-the-art downstream summarization model. Our results show that PURETEXT can greatly improve upon downstream model baselines for multiple datasets and models. It excels at improving summarization for long articles. We hypothesize that PURETEXT is particularly effective on long articles because truncation of these articles often results in removing important sentences. This suggests that it is most applicable to datasets similar to WikiHow and Reddit, where key sentences are evenly distributed throughout each article. Conversely, journalistic articles tend to have important sentences concentrated towards the beginning of the article, making it less effective. We encourage future work to expand on the comprehensiveness of our study and to continue exploring the dataset- and model-agnostic nature of such a sentence filtering layer for downstream summarization.

References

Ahsaas Bajaj, Pavitra Dangati, Kalpesh Krishna, Pradhiksha Ashok Kumar, Rheeya Uppaal, Brad Windsor, Eliot Brenner, Dominic Dotterrer, R. Das, and A. McCallum. 2021. Long document summarization in a low resource setting using pretrained language models. In *ACL*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *International Conference on Learning Representations*.

Alexios Gidiotis and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of academic articles. *ArXiv*, abs/2004.06190.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

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, Volume 1 (Long and Short Papers)*, pages 2519–2531, Minneapolis, Minnesota. Association for Computational Linguistics.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

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

Yang Liu. 2019. Fine-tune bert for extractive summarization.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar GuÌ‡lçehre, and Bing Xiang. 2016. 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.

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.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Shuai Wang, Xiang Zhao, Bo Li, Bin Ge, and Daquan Tang. 2017. Integrating extractive and abstractive models for long text summarization. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 305–312. IEEE.

Hao Xu, Yanan Cao, Ruipeng Jia, Yanbing Liu, and Jianlong Tan. 2018. Sequence generative adversarial network for long text summarization. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 242–248. IEEE.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Haoyu Zhang, Jingjing Cai, Jianjun Xu, and Ji Wang. 2019. Pretraining-based natural language generation for text summarization. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 789–797, Hong Kong, China. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia. Association for Computational Linguistics.