Abstractive Summarization of Social Media Using Multiple Transformer Stages

Skyler Roh

University of California, Berkeley
skylerroh@ischool.berkeley.edu

August 1, 2020

Abstract

Abstractive summarization has made substantial progress in recent years due to the introduction of transformers along with various self-supervised pretraining techiniques. While the majority of research in this area has been performed on news article datasets, other forms of text have not been widely tested. Other mediums such as posts on the web forum Reddit are significantly less structured. In this paper, a hierarchical model composed of a transformer-based sentence encoder and additional multihead-attention for sentence selection is proposed to better retain salient information while remaining within sequence length limits imposed by transformer architectures.

1. Introduction

Automatic summarization has largely surrounded news article datasets. These articles are short distinct sets of information on current events and the language used is relatively standardized and formal, consisting of primarily of extractive highlights as summaries and much of the relevant information being contained within the first couple sentences. Comparatively, the informal nature of internet forums such as Reddit offer far less structure and cover a wide range of topics. Transformer networks such as BERT [Devlin et al.(2018)Devlin, Chang, Lee, and Toutanova] have made significant progress towards accounting for sense discrepancies with the use of self attention and this work will expand on the use of hierarchical transformer architectures and finetuned PEGASUS [Zhang et al.(2019)Zhang, Zhao, Saleh, and Liu] to create abstractive summaries of the 50 most common subreddits found in the tl;dr Reddit dataset [Völske et al.(2017)Völske, Potthast, Syed, and Stein]. This dataset differs significantly from datasets commonly used in summarization training such as CNN/DailyMail and Gigaword in nature, news vs. interest forums, diversity of language used both in vocabulary and formality. In this research, I address differences in the position of salient text within the document. Compared to single document news summaries, which present highlights mainly at the introduction of the text, other mediums such as Reddit posts may present information relevant information both in the introduction–setting context or asking a question–and at their conclusion–a punchline or a concluding position. While transformer based models have significantly furthered capabilities in language understanding, models such as BERT and it's variants have token limits that require text to be shorter than a certain length. A common technique to satisfy this constraint is to do a truncation after the token limit. The primary contribution of this

work describes the use of a sequential series of transformers to encode and rank shorter sequences of the text to remove the least relevant information then create abstractive summaries via fine-tuned PEGASUS model. This method produces better summaries than a pure encoder-decoder transformer by first producing recall focused extractive summaries that preserves relevant text better than truncation methods.

2. Related Work

1. Extractive Summarization

The goal of extractive summarization is to select appropriate sentences or snippets of source text that are most salient to the corresponding document which maximize metrics such as ROUGE and human evaluation. Recently, there have been various efforts to finetune BERT to the task of extractive summarization. BERTSUM [Liu(2019)] fine-tunes BERT to accept a sequence of input sentences along with interval segment embeddings to distinguish sentence position and output sentence scores. HIBERT [Zhang et al.(2019)Zhang, Wei, and Zhou] proposes a differing architecture and pretraining technique that first encodes each sentence independently then pretrains on masked sentence task. The architecture proposed by HIBERT will be used for the initial extractive task of compressing the original post as the independent sentence encoding allows for longer texts to be input before ranking sequences within the document.

2. Abstractive Summarization

Abstractive summarization takes the form of sequence to sequence modeling as input text is encoded into a hidden vector state that is then decoded to generate new text. Various methods have been developed over recent years such as Pointer-Generator Networks [See et al.(2017)See, Liu, and Manning] which probabilistically choose between extracting source text and generating new text with LSTMs and attention and PEGASUS, a transformer-based encoder-decoder. PEGASUS pretrained on gap sentence generation on C4 dataset and fine-tuned on smaller downstream datasets for abstractive summarization has produced SOTA results on nearly all major news article-summary datasets across all ROUGE metrics; however, like other transformer models, PEGASUS has input sequence length limits in addition to the computation cost of self-attention layers being quadratic in the sequence length.

3. Hierarchical Models

In order to narrow the focus of the abstractive summarization task, various hierarchical approaches have utilized two model stages, one which first extracts salient sentences which are then run through sequence-to-sequence models for abstraction. One such model is Hi-MAP [Fabbri et al.(2019)Fabbri, Li, She, Li, and Radev], which tackles multi-document summarization by introducing sentence-level MMR scores to a Pointer-Generator network for summarization. Tangentially, some models combine additional inputs with encoder outputs to then be decoded. One such model that has held SOTA results on the Reddit tl;dr dataset is VAE-PGN [Choi et al.(2019)Choi, Ravuru, Dryjański, Rye, Lee, Lee, and Hwar which combines the output of a variational autoencoder with the encoding of a Pointer-Generator network before decoding.

3. Methods

1. Dataset

The Reddit tl;dr dataset consists of over 3.8M content-summary pairs. This dataset is a normalized mix of original posts and comments. When examining the dataset for this research project, texts that were missing the subreddit or title were excluded, leaving 912k remaining. From these, up to 2000 for each of the top 50 occuring subreddits were utilized for training (82k total) and a mutually exclusive set of up to 100 from each subreddit (4.4k total) were

utilized for validation and test respectively. For each example, the title and subreddit were affixed to the beginning of the content text. In addition to the tl;dr section of the post being used as the target for the abstractive summarization task, labels needed to be generated for each sentence for the extractive summarization task. This was accomplished using a greedy approach outline by Liu in BERTSUM and Nallapati in SummaRuNNer [Nallapati et al.(2016)Nallapati, Zhai, and Zhou] which incrementally gives positive labels to up to n sentences that maximize ROUGE f1 metrics against the target text. The labels used for training in the model outlined below utilize the same greedy labeling approach, however, the task optimizes for recall instead of f1 in order to preserve relevant information at some cost to keeping some extra non relevant text.

2. Model Architecture

The methods described in this section outline the procedure for ranking sentences within a source text utilizing pretrained DistilBert [Sanh et al.(2019)Sanh, Debut, Chaumond, and Wolf] for sentence encoding and an additional multihead attention layer that is optimized for recall ROUGE metrics followed by finetuning PEGASUS for abstractive summarization. Each of these models was trained on a few epochs of the training data 6 and 2 for the extractive and abstractive portions respectively using an AdamW optimizer with warmup.

- Let D = (S1, S2, . . . , S|D|) denote a document, where Si = (wi1, wi2, . . . , wi|Si|) is a sentence in D and wij is a word in Si.
- Encode each Si in D with pretrained DistilBERT as the average encoding of all wij.
- Pass each Si encoding as sequence into layer of multihead attention.
- Final sentence encodings run through feed forward layer to estimate probability of saliency.
- Rank and select up to top-k (20) sentences from document and reconstruct text in chronological order.

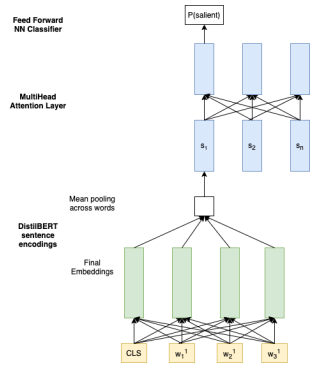

Figure 1: *Diagram of extractive sentence selection model*

- Finetune PEGASUS on pairs of extracted text and target summaries.
- Evaluate on ROUGE f1 metrics (r1, r2, and rL).

4. Results and Analysis

1. Baselines

To test the hypothesis that Reddit posts do not contain all salient information at the beginning of the text, a series of baseline metrics were collected by considering the starting k (k=128) tokens, ending k tokens, and bookend tokens (first k/2 and last k/2) tokens as summaries to evaluate against target summaries. The following baselines were examined across each subreddit in the dataset and as a total aggregate. Figure 2 below displays a comparison of the ROUGE-2 scores of the starting-k tokens compared to the ending-k and bookend-k counterparts. In the evaluation of all posts, the bookend-k token extraction outperformed the other two with ROUGE scores of 19.1/4.8/12.1 followed by starting-k with 18.2/4.3/11.5 and ending-k following just below that. Further inspection of subreddits that exhibited the greatest gain when using the bookend baseline compared to the starting tokens revealed that each of these subreddits can be generally categorized under advice ('relationships', 'relationship_advice', 'dating_advice', and 'advice'). This pattern seems natural as often such posts are longer, with average lengths between 460

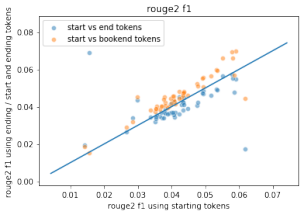

Figure 2: *Comparing baseline metrics for summaries starting-k tokens vs ending-k and bookend-k (first k/2 and last k/2) for each subreddit. Line denotes metric is equal between each baseline. As seen, the bookend-k model outperforms both the starting-k for nearly all subreddits.*

and 590 tokens, and concluding with final takeaway conclusions following a body of context.

2. Sentence Extraction

Two extractive model architectures were tested. The main difference between the two was the absence or presence of a intermediate multihead-attention layer between the sentence encodings from DistilBERT and feed forward dense layers for classification. The best loss and recall @ top-k performance was found when utilizing a multihead-attention layer with positional embeddings followed by a single dense layer.

The amount of distillation between the source text and the extracted text from top-k sentences depends greatly on the choice of k. Figure 5 visualizes the distribution of text

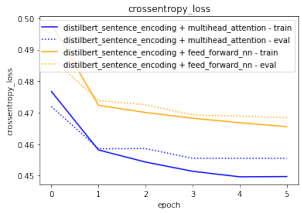

Figure 3: *Distribution of post lengths in number of wordpiece tokens. Selecting top 20 sentences predominantly compresses the texts that are greater than 512 tokens closer to this sequence limit. Meanwhile selecting top 10 ranking sentences reduces length of all documents to 400 tokens or less*

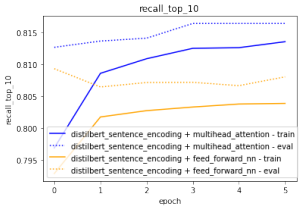

Figure 4: *Distribution of post lengths in number of wordpiece tokens. Selecting top 20 sentences predominantly compresses the texts that are greater than 512 tokens closer to this sequence limit. Meanwhile selecting top 10 ranking sentences reduces length of all documents to 400 tokens or less*

lengths and the table below describes the summary metrics of for the original text and the extraction of top-k sentences for k=10 and k=20.

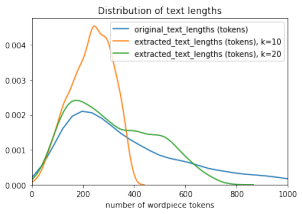

Figure 5: *Distribution of post lengths in number of wordpiece tokens.*

Table 1: *Document Length Statistics (tokens)*

k	Mean	StDev	Max
Original (k = all)	438.9	386.9	4487
20	315.8	165.6	767
10	231.6	81.7	390

3. Abstractive Summaries

The finetuning of PEGASUS was trained on 2 epochs (50k steps and batch_size=4)[1] of each of the following outputs from the best performing extractive model: the original text, top-10 ranked sentences, and top-20 sentences. Each of these inputs was truncated at 512

[1] training length limited due to computation restrictions, running on n1-highmem-8 with NVIDIA TESLA T4

tokens and max summary length of 128 tokens. Evaluation on the summaries generated from using the top-20 showed the best performance with slightly higher values on all ROUGE scores. These results suggest that the extractive sentence selection in longer texts to fit within the transformer sequence length limit does perform better on this Reddit tl;dr task than doing a simple truncation; however, doing minimal content removal (k=20) did perform better than the more selective (k=10) probably due to less information loss. The largest gains are seen on rouge2 metrics. Each of the finetuned PEGASUS models produce ROUGE metrics that beat SOTA presented in Towards Summarization for Social Media [Syed et al.(2019)Syed, Völske, Lipka, Stein, Schütze, and Potthast][2].

Table 2: *ROUGE(f) scores after finetuning PEGASUS for 2 epochs along with scores presented in Towards Summarization for Social Media (Syed et al. 2019)*

Model(k)	r1	r2	rL	bleu
Original	21.7	7.3	17.6	4.17
20	**21.9**	**7.7**	**17.8**	4.2
10	21.6	7.5	17.7	4.21
tldr-bottom-up	20	4	15	_
transf-seq2seq	19	5	14	_
unified-vae-pgn	19	4	15	_

5. Conclusion

The gap sentence generation pretraining technique introduced by PEGASUS has greatly advanced the state of abstractive summarization, beating previous SOTA pointer-generator models and other seq2seq transformers. The introduction of a hierarchical method of distilling longer texts to fit into the length limit characteristic of transformers by utilizing DistilBERT encodings of sentences followed by multihead attention layers does offer marginal increase in performance; however, there are many areas for further improvement. The method for labeling salient sentences at the extractive model layer seems particularly pertinent both in the greedy nature of the selection used and the choice of maximizing ROUGE recall scores. This choice was made in effort to maximize retention of potentially relevant information. Alternatively, the exploration of a less greedy algorithm and f1-based scores or utilizing networks with similar architectures but pretrained on different self-supervision tasks like HIBERT if available may prove to introduce less noise in sentence ranking. Furthermore, longer finetuning could reveal more significant deviations in overall performance. Lastly, human evaluation of summaries revealed a model tendencies such as copying quotes verbatim. Similar to results from Zhang et al's research in PEGASUS, abstractive seq2seq models still copy text far more often than human generated summaries.[3]

[2]Scores presented in Towards Summarization for Social Media (Syed et al. 2019) based on test set no longer available for scoring

References

[Choi et al.(2019)Choi, Ravuru, Dryjański, Rye, Lee, Lee, and Hwan
Hyungtak Choi, Lohith Ravuru, Tomasz Dryjański, Sunghan Rye, Donghyun Lee, Hojung Lee, and Inchul Hwang. 2019. VAE-PGN based abstractive model in multi-stage architecture for text summarization. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 510–515, Tokyo, Japan. Association for Computational Linguistics.

[Devlin et al.(2018)Devlin, Chang, Lee, and Toutanova]
Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[Fabbri et al.(2019)Fabbri, Li, She, Li, and Radev]
Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive

[3]see github repo https://github.com/skylerroh/tldr-reddit-summarization for example summaries

hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

[Liu(2019)] Yang Liu. 2019. Fine-tune BERT for extractive summarization. *CoRR*, abs/1903.10318.

[Nallapati et al.(2016)Nallapati, Zhai, and Zhou] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2016. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *CoRR*, abs/1611.04230.

[Sanh et al.(2019)Sanh, Debut, Chaumond, and Wolf] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv e-prints*, page arXiv:1910.01108.

[See et al.(2017)See, Liu, and Manning] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368.

[Syed et al.(2019)Syed, Völske, Lipka, Stein, Schütze, and Potthast] Shahbaz Syed, Michael Völske, Nedim Lipka, Benno Stein, Hinrich Schütze, and Martin Potthast. 2019. Towards summarization for social media - results of the TL;DR challenge. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 523–528, Tokyo, Japan. Association for Computational Linguistics.

[Völske et al.(2017)Völske, Potthast, Syed, and Stein] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.

[Zhang et al.(2019)Zhang, Zhao, Saleh, and Liu] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *arXiv e-prints*, page arXiv:1912.08777.

[Zhang et al.(2019)Zhang, Wei, and Zhou] Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.