XLNetSumExtAbs, Text Summarization with Pretrained Encoders

Brandon Castaing, Aditi Das
University of California, Berkeley
{bcastaing,adas}@berkeley.edu

Abstract

Automatic text summarization is a rapidly evolving application of natural language processing (NLP) as a result of the research and industry demands for succinct, comprehensible, and accurate summaries from large texts grows. Recently XLNet addressed a couple of BERT's limitations resulting in the latest state of the art performance on several NLP tasks, but did not achieve this same status in abstract text summarization [1]. PreSumm, a simple fine-tuned model based on BERT achieved the state of the art in abstract text summarization by outperforming the previous best-performed systems on extractive as well as abstractive summarization [2]. We explore the potential of using the XLNet model with the PreSumm finetuning layer for both extractive and abstractive summarization. Even with limited computational resources, our experiments yield encouraging summarization results evaluated using several ROUGE metrics.

1 Introduction

Pretrained language models have taken the natural language processing world by storm, and we seek to explore the potential of the latest groundbreaking works of XLNet and PreSumm in combination. Currently, there are two main approaches to automatic text summarization: extractive and abstractive. Extractive summarization creates summaries with verbatim words from the original document while abstractive summarization produces new words that make the summary more human-like. The previous state of the art algorithm, BERT, exhibited two major drawbacks which XLNet addressed to become the state of the art pretrained language model by achieving empirically superior results in several NLP tasks. Most notably, text summarization was not included in this task achievement list though. Within a month of XLNet's publication, PreSumm, a simple fine-tuned model based on BERT achieved the state-of-the-art performance on extractive and abstractive summarization. In this paper, we explore the potential of using components of the XLNet model with the PreSumm finetuning layer for both extractive and abstractive summarization. Our experiments are run on the CNN / Daily Mail dataset using the ROUGE-1, ROUGE-2, and ROUGE-L metrics, then compared with BERTSUM as a benchmark [5].

2 Background

The formulation of our experiment is inspired by earlier research on abstract text summarization in natural language processing. Liu et al. fine-tuned the previous state of the art pretrained language model BERT by adding a series of encoder/decoder optimization layers [2]. Their novel approach to build an extractive summarization model and leverage the resulting model for a second model to perform abstract summarization boosted the quality of the model's generated summaries. Their resulting abstract text summarization model BertSumExtAbs achieved state of the art results in its specific task. Yang et al. proposed the pretrained language model XLNet which addresses BERT's shortcomings in data corruption for training via masking and token independence assumption [1]. The authors were able to improve upon BERT by leveraging the best of both an autoregressive model similar to Transformer XL and autoencoding model similar to BERT [4]. This unprecedented pretrained language model achieved state of the art results in 18 natural language processing tasks, and remarkably bested BERT in 20 tasks. Notably, text

summarization was not among the list of tasks though.

3 Methods and Contributions

3.1 Incorporating Components from XLNet

summarization was not among the list of tasks though.

Given XLNet's superior performance in multiple natural language processing tasks and resolution of BERT's shortcomings, we theorized that an abstract text summarization neural network model based on XLNet would empirically outperform one built with BERT. We integrate two key components from XLNet into our resulting model, namely the XLNet-base transformer layer and XLNet tokenizer. The most notable components of the XLNet-base transformer layer are two attention streams, one for query and content representation respectively. These attention mechanisms help the language model to retain context over several tokens. Multiple XLNet-base transformer layers are stacked to serve as the initial layers of our deep neural network architecture for extracting document level features. The XLNet tokenizer uses a SentencePiece model to create a dense representation of subword units of a specified size [6]. It is used for tokenizing the source document prior to being fed through the model as well as for decoding the resulting target text output from the model. We start with the PreSumm framework and implement the above modifications to replace the BERT language model components with XLNet.

3.2 Fine-tuning XLNet for Summarization

Following the pattern of PreSumm, we used XLNet word embeddings along with interval segmentation embeddings to distinguish each sentence within a document. For the extractive summarization version of our model XLNetSumExt, we added a <CLS> token before the start of each sentence and a <SEP> token after each sentence to capture the representation for each sentence. After obtaining the sentence vectors from XLNet, we stack a couple of inter sentence Transformer layers to capture document level features for extracting sentences. For each sentence, we calculate the final predicted score. Cross entropy loss is used to calculate the final predicted score against the target label. For the abstractive summarization version of our model XLNetSumExtAbs, we use the fine-tuning transformer-decoder framework from BertSumExtAbs for abstractive summarization on top of XLNet. The output of the XLNet layers are passed to a 6-layered Transformer decoder network that are initialized randomly. Similar to BertSumExtAbs, we use an Adam optimizer, warm up-steps and the same learning rate for our model. Finally, we took the PreSumm framework and implemented adjustments to the decoding logic to support XLNet's unique word embeddings.

3.3 Dataset and Preprocessing

The CNN / Daily Mail dataset is used for training our summarization models [5]. We used the standard data splits as seen in Liu et al. for training, validation, and testing the dataset resulting in 90,266/1,220/1,093 CNN documents and 196,961/12,148/10,397 DailyMail documents [2]. We first split sentences with the Stanford CoreNLP toolkit, and pre-processed the dataset using a modified PreSumm pipeline to leverage XLNet's tokenizer and its unique SentencePiece model component. All input documents were truncated to 512 tokens to fit the layer dimensions of XLNet.

The dataset contains abstract gold summaries, which are not readily suited to training extractive summarization models. A greedy algorithm was used to generate an oracle summary for each document. The algorithm greedily selects sentences which can maximize the ROUGE scores as the oracle sentences. We assigned label 1 to sentences selected in the oracle summary and 0 otherwise.

As previously mentioned, we used XLNet's tokenizer to convert our text into tokens that correspond to XLNet's vocabulary. We maintained usage of the BERT token pattern:

[CLS] Sentence one [SEP][CLS] Sentence two [SEP]

For each tokenized input sentence, we needed to create:

- input ids: a sequence of integers identifying each input token to its index number in the XLNet tokenizer vocabulary

- segment mask: a sequence of 1s and 0s used to identify whether the input is one sentence or two sentences long. For one sentence inputs, this is simply a sequence of 0s. For two sentence inputs, there is a 0 for each to-

Figure 1: Illustration of the input and encoder layer architecture for XLNetSumExtAbs.

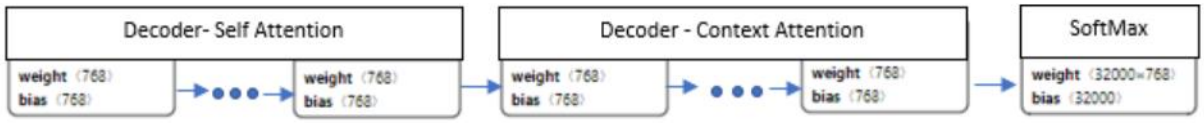

Figure 2: Illustration of the decoder and output layer architecture for XLNetSumExtAbs.

ken of the first sentence, followed by a 1 for each to-ken of the second sentence.

- position mask: position of each token.
- labels: a single value of 1 or 0. In our task 1 means sentence is part of the summary, 0 otherwise.

3.4 Architecture: XLNetSumExtAbs

Our final model architecture consists of XLNet word embeddings being fed through 12-XLNet-base layers followed by 6-BertSum decoder fine tuning layers. As seen in Figure 1, the word embeddings feed directly into the first XLNet-base layer and would continue through eleven more similar layers until it reaches the decoder layers. The XLNet layers are made up of 8 bi-directional attention heads with an intermediate feed forward hidden size of 3072 nodes. As previously seen in PreSumm, we first create an extractive summarization model with 12-XLNet layers fed into 2-PreSumm encoder layers. The updated 12-XLNet layer weights are used as starting weights for the XLNet layers during abstract summarization training. As seen in Figure 2, the resulting output from the last XLNet-layer would feed into the decoder layer and continue through five more similar layers until it reaches the final SoftMax output layer which generates the resulting summary's word vector. Our model has similar architecture hyperparameters to the XLNet-base language model and PreSumm's fine tuning layers, and results in similar layer sizes for the language modeling and fine-tuning portion of XLNetSumExtAbs respectively.

3.5 Implementation: XLNetSumExtAbs

We train XLNetSumExt on 2 Tesla V100 GPU chips for 50K steps by minimizing cross-entropy loss using 2 Adam optimizers with drop out of 0.1 for regularization, linear learning rate decay of .002, accumulation of 25, and a batch size of 512. This configuration took about one and a half days for training to complete. Moreover, we trained XLNetSumExtAbs leveraging the aforementioned XLNetSumExt model as a base and using the same configuration previously specified. This final training took about 6.5 days to complete. Furthermore, we attempted to train a variant of XLNetSumExtAbs with XLNet-Large as the base pretrained model. This resulted in size incompatibilities with the decoder layers responsible for abstractive summarization.

4 Results and Analysis

Model checkpoints were saved and evaluated on the validation set every 1,000 steps. We selected the top-3 checkpoints based on the evaluation loss on the validation set and report the averaged results on the test set. On CNN/Daily Mail dataset, we report the full-length F1 score of the ROUGE-1(unigram overlaps), ROUGE-2 (bigram overlaps) and ROUGE-L metrics (subsequence overlaps), calculated using the PyRouge package. ROUGE is not an ideal evaluation metric due to its reliance on the same lexicon being used for the source and target which is especially problematic for abstract summarization. It is the industry standard metric, so we elected to use these metrics as a good first approximation of our model's performance.

MODEL	ROUGE-1	ROUGE-2	ROUGE-L
Extractive			
BERTSUMEXT	43.35	20.25	39.68
XLNETSUMEXT	42.43	19.33	38.84
Abstractive			
BERTSUMEXTABS	42.16	19.49	39.16
XLNETSUMEXTABS	27.04	7.03	24.14

Table 1: ROUGE F1 results for various models on the CNN/Daily Mail dataset

As seen in Table 1, we compare the performance of PreSumm's models with our XLNet versions against the CNN / Daily Mail dataset. We classify them into two groups based on whether they are extractive or abstract models. Extractive summarization with XLNetSumExt performs competitively with BertSumExt with less than a difference of one in all three ROUGE metrics. Surprisingly, abstractive summarization with XLNetSumExtAbs is substantially worse than BertSumExtAbs. We speculate that a couple key errors may be responsible for our poor performance in abstract summarization, but due to time constraints were not able to validate these hypotheses. First, BERT is a masked language model where only the masked tokens are predicted. This is a significant difference with XLNet's permutation language modeling since it uses all tokens when predicting in a random order. One of the mistakes we have done is we left the masking on when we trained the XLNetSumExtAbs model. Second, XLNet's token pattern is:

Sentence_A + [SEP] + Sentence_B + [SEP] + [CLS]

where [CLS] is at the end instead of the beginning of each sentence. We kept the Bert pattern where [CLS] is at the beginning of each sentence while training our model. This token pattern may have been more critical to model training than originally anticipated. Additionally, we recognize that we were not able to fully leverage our dataset as we were only able to process 50k steps with the allotted time and computational resources. It would have been ideal to have run at least one epoch. We managed with our limited computational resources from our laptops and educational cloud credits to train a few hyper parameter combinations and would like to explore further hyper parameter optimization as well.

5 Conclusion

We explored the potential of using XLNet with PreSumm's finetuning layer for extractive and abstractive text summarization respectively. Although our results are not ground-breaking, it would be interesting to further test XLNetSumExtAbs with the changes mentioned in the previous section. We hope that our methods and results will compel fellow colleagues in the academic and professional world to continue our experimentation.

Acknowledgements

The authors would like to thank Mark Butler and James Kunz from the School of Information at the University of California, Berkeley for their advice and mentorship during the development of this project. We commend the excellent work of the authors of the XLNet and BertSum papers who progressed the field of natural language processing with their recent papers.

References

[1] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. and Le, Q. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. [online] arXiv.org. Available at: https://arxiv.org/abs/1906.08237 [Accessed 25 Sep. 2019].

[2] Liu, Y. and Lapata, M. (2019). Text Summarization with Pretrained Encoders. [online] arXiv.org. Available at: https://arxiv.org/abs/1908.08345 [Accessed 25 Sep. 2019].

[3] Devlin, J., Chang, M., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. [online] arXiv.org. Available at: https://arxiv.org/abs/1810.04805 [Accessed 2 Dec. 2019].

[4] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, Ruslan Salakhutdinov. (2019). Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. [online] arXiv.org. Available at: https://arxiv.org/pdf/1901.02860.pdf [Accessed 2 Dec. 2019].

[5] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, Phil Blunsom. (2015). Teaching Machines to Read and Comprehend. [online] arXiv.org. Available at:

https://arxiv.org/pdf/1506.03340.pdf [Accessed 2 Dec. 2019].

[6] Taku Kudo, John Richardson. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. [online] arXiv.org. Available at: https://www.aclweb.org/anthology/D18-2012.pdf [Accessed 2 Dec. 2019].

A Sample Summaries

D(Source document): Much like coconut oil, pumpkin seed oil is thought to help boost shiny locks. Pumpkin seeds are rich in vitamins A, K and E, as well as vital minerals and fatty acids, which can help strengthen hair and even help boost hair growth. . . . It also apparently has similar skin-boosting benefits. With high amounts of polyunsaturated fatty acids and natural antioxidants, pumpkin seed oil helps promote normal cell structure and retains moisture, which ensures skin is hydrated and youthful. Zinc and vitamin E from the pumpkin seed can also improve the skin's healing process, helping to fight off acne, beat scarring and maintain skin renewal.
O(Original summary): Experts say it's time to replace coconut oil with pumpkin seed oil.Rich in vitamins A, K and E, as well as vital minerals and fatty acids.
G(Abstract summary using BertSumExtAbs): health gurus are touting pumpkin seed oil as the next big thing.it can help enhance your mood,renew skin and even reduce menopause symptoms.powerhealth,who supply the health produce , say it hydrates hair follicles giving shiny and lustrous strands
G(Abstract summary using XLNetSumExtAbs): Studies have even found that pumpkin seeds are a mood-boosting food.with high amounts of polyunsaturated fatty acids and natural antioxidants , pumpkin seed oil helps promote normal cell structure and retains moisture , which ensures skin is hydrated and youthful.

B Github Repository

https://github.com/brandon-castaing-ucb/XLNetExtAbsSum