Comparing Abstractive and Extractive Summarization Techniques Through Question Answering Evaluation

Vishal Tien, Sanjana Vasudevan, Andrew Clark
Department of Electrical and Systems Engineering
University of Pennsylvania
vtien@seas.upenn.edu, svas@seas.upenn.edu, clarkand@seas.upenn.edu
https://github.com/vtien/abstractive_extractive_summary_evaluation

Abstract

Deep learning models for text summarization have seen significant improvements over the past few years, as typically evaluated by the common ROUGE metric. However, it has been found that ROUGE scores do not always correlate with human evaluation of summaries. As such, alternative metrics to ROUGE are being explored, such as question-answering based analysis. The quality and effectiveness of summaries can be judged effectively by question-answering ability, as question-answering provides context on how well summaries capture information from a text that is deemed to be important, as defined by the questions. In this work we present an adaptation of Bahdanau et al.'s 2015 neural machine translation model for abstractive text summarization of CNN news articles. In addition to adapting the model for abstract text summarization, we introduce various improvements to the model, such as beam search and label-smoothing, to improve our output summaries. We further implement an extractive summarization model that utilizes Google's BERT framework in order to benchmark the abstractive model. We then evaluate and compare these abstractive and extractive summarization models using ROGUE scores and question answering scores via the Hugging Face Transformers framework.

1 Introduction

Effective text summarization techniques have far-reaching applications, including legal document analysis, customer reviews summarization, and financial report information extraction to name a few. Text summarization promises to save time and resources in many domains. Extensive work has been done in the extractive text summarization domain, and extractive text summarization tasks models that utilize Google's BERT framework have been used to achieve state of the art performance on lecture summarization tasks [1]. While effective extractive summarization is indeed useful in its own right, much of the utility of natural language understanding lies in a specific framework's ability to effectively reason over textual representations of real world data. In this work, these representations are summaries of a source text. As such, while ROUGE scores are often used to judge summarization model outputs, question-answering as a metric of summary evaluation promises to yield more utility in the form of an effective judgement of a model's reasoning ability. Taking this logic one step further, abstractive text summarization models promise to complete a more involved task then the one completed by extractive models. Namely abstractive models aim to create novel, coherent sentences via context-awareness and actual understanding of the language present in the source text. Conversely, extractive text summarization methods create a predicted summary by piecing together in-tact key phrases or sentences without creating any novel text [2]. With this difference between extractive and abstractive deep learning models in mind, we hypothesize that abstractive text summarization models will be able to capture the overarching message of a source text in a

summary more holistically than a pure extractive model. If done effectively, the ability of abstractive text summarization models to capture underlying meaning will in turn allow an abstractive model to perform better on question-answering evaluation tasks that test reasoning ability, compared to an extractive model, while still achieving comparable ROUGE scores. We investigate this hypothesis here as a method for comparing two deep learning text summarization approaches with respect to not only ROUGE scores, but also the more novel question-answer based metric.

2 Contributions

In this work, we take a baseline abstractive summarization model, improve upon it by making modifications, and then compare the initial and improved model to a state-of-the-art extractive summarization model using ROUGE scores and a novel question-answering evaluation approach. More specifically, we present an adaptation of the model in Bahdanau et al.'s 2015 paper "Neural Machine Translation by Jointly Learning to Align and Translate" to create abstractive text summaries of CNN news articles. In addition to adapting the Bahdanau model for abstractive text summarization, we introduce a model utilizing beam search and label smoothing regularization, and one with an adaptive learning rate in an attempt to improve our model's ability to produce effective summaries. We devised a beam search decoding method applied to our specific model from scratch, and adapted a label smoothing class sourced from [5]. By utilizing the Huggingface Framework, we compare the question-answering ability of abstractive model's predicted summaries to the summaries of a baseline extractive text summarization model, namely Google's BERT framework for extractive text summarization, through manually crafted questions. We further benchmark our results by ROUGE score, finding that ROUGE-1 score is correlated with mean QA accuracy, but not with question answering confidence.

3 Background

As mentioned above, there are two main approaches to automatic document summarization: extractive and abstractive. Extractive summarization involves extracting key words and phrases from the source text and combining them to capture the essential information of the text. One of the most prominent being the extractive model BERT. BERT features a transformer model combined with self-supervised pre-training, which is extremely effective when tuned to a specific task [2]. In contrast, abstractive summarization involves the potential addition of words and phrases that are not in the source text to the summary. The most popular method for summarization is the use of seq2seq models which map input sequences to output sequences. These usually involve the use of RNN encoder/decoders [3]. One difficulty in the summarization task is evaluation. It is difficult to objectively quantify the quality of a summary or compare quality between summary. The most prominent metric for summarization tasks is the ROUGE score which is a measure of n-gram overlap between a summary and a target summary [4]. However, due to interest in the reading comprehension field, question answering has been posed as an alternative to measure if the most important information in a summary is retained. The CNN/DailyMail dataset is the most widely used primary dataset for summarization tasks. This dataset contains 1,000,000+ news articles, on average 30 sentences long, from CNN and DailyMail along with several summary sentences which contain the most important information in the article. These can be used as a target summary, of which we use a subset for our task [6].

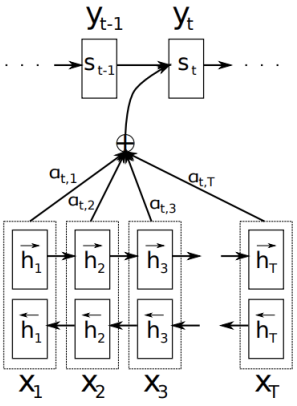
Question answering is part of the larger field of the reading comprehension task in NLP. Question answering can be cloze-domain, which involves filling in a blank of an excerpt with a word in the corpus, or open-domain, which involves asking an arbitrary question and getting the best response from the text source. [7] A big difficulty in question answering, similar to summarization, is the creation of datasets and the evaluation of question answering ability. Hugging Face is an NLP Python package which has Extractive Question Answering functionality as used in this paper. The Hugging Face backend is based on a BERT extractive summarization model, pretrained on the SQuAd question answering. This model can then be fine tuned on any data of your choosing. Hugging Face, when given a summary and a question, returns the proposed answer as well as the start and end indices of the answer and a confidence score of the answer.

4 Related Work

In 2015, Rush et al. first used an end-to-end neural network for summarization, consisting of a seq2seq model with an attention mechanism introduced by Bahdanau, et al.[4][1] See et al. successfully introduced using pointer-generator networks along with the previous LSTM encoder/decoder for text summarization [8]. In Just News It: Abstractive Text Summarization with a Pointer-Generator Transformer, Vasavada and Bucquet build on the work of See et al. by introducing a transpointer, a combination of a transformer and pointer generator network [9]. Beam search was introduced by Wu et al. in Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation as an aspect for the similar NLP task of automated translation [10]. Vaswani, et al introduced the successful and influential transformer architecture, also using beam search, in Attention is All You Need [11]. In 2015, Hermann et al. released their work on developing the CNN/DailyMail dataset for use on question answering tasks as well as attention-based LSTM networks for these tasks. [5] Eyal et al. developed the Answering Performance for Evaluating Summaries (APES) metric in Question Answering as an Automatic Evaluation Metric for News Article Summarization [12]. This involved evaluating summaries of the CNN/DailyMail dataset by scoring their ability to fill in a blank word in the target summary. The base QA model used for APES, an attentive GRU encoding architecture, was from Chen, et al.'s A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task [13].

5 Approach

Our approach first involved replicating recent models in both abstractive and extractive summarization fields of work. We decided to pick an encoder / decoder model with attention as our baseline abstractive model, as it represents one of the most effective abstractive model architectures that newer models attempt to improve upon. As this was our group's first exposure to NLP in the field of summarization, replicating this model architecture provided a good foundation for understanding current state-of-the-art models, such as the newer transformer abstractive models. We leave it for future work to extend the results found here to transformer architectures and other recent developments to attention based encoder / decoder models (pointer generator networks, copy mechanisms, etc.). To benchmark the results achieved by our abstractive model, we compare them with a pre-trained BERT extractive summarization model developed by Google [2]. For our abstractive model, we follow the encoder/decoder structure as described in Bahdanau et al. [1].



Our first step was to find a working implementation of this model architecture in PyTorch [14]. The model is constructed with a bi-directional GRU for the encoder, which follows the simple recursive formula shown below from left-to-right and right-to-left.

$$\mathbf{h}_j = GRU(x_j, \mathbf{h}_{j-1}) \tag{1}$$

The bi-directional GRU effectively encapsulates the entire source article in context by concatenating the hidden states from both directions. The decoder is a conditional GRU, which follows a similar recursive formula as the encoder, except it takes in an additional input ci, the context vector, which constitutes one improvement upon previous encoder / decoder models. The Bahdanau attention

mechanism is implemented to dynamically select part of the source sentence that is most relevant for predicting the current target word, creating this context vector ci.

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{2}$$

After taking in the hidden state from the encoder, the previously generated word, and the context vector as inputs, a linear layer and a softmax is applied to output a probability distribution of the predicted word for this time step. The Bahdanau encoder/decoder model was first developed for neural machine translation. We adapted this model to our summarization task by modifying various hyperparameters. The model's ability to work following a few parameter changes highlights the ability of neural network models to adapt to other NLP tasks. The models were evaluated using question answering via the Hugging Face Transformers Python package. We wanted to manually write questions so that we had complete control over the content of the questions, allowing us to write questions that captured the most salient information in the article. We crafted three questions per article and recorded the most probable answers for these questions on a given summary and confidence scores for these answers as determined by the Hugging Face model. Due to the time-consuming nature of manually developing questions for each article, we used 50 articles for this evaluation.

6 Experimental Results

We manually performed initial preprocessing of the data, which can be found in the pre-processing notebook in our github. Due to resource constraints, we filtered the training set to only consist of the 55,000 shortest articles of text (fewer than 400 words). We then used torchtext to tokenize, pad, and add start and end tokens to our data. We used Google Colab Pro to train our models by taking advantage of the GPU accelerator. During training, we monitored both validation loss and validation perplexity.

Target Summary: andrea pirlo scored a stunning free - kick for juventus against torino . however , juve fell to a 2 - 1 defeat and missed the chance to wrap up their fourth consecutive serie a title . despite the loss , juve remain 14 points clear at the top of the league table . pirlo is still playing at the top of his game at 35-years - old .
BERT: Andrea Pirlo scored a stunning free-kick for Juventus, but couldn't stop his side from sinking to a 2-1 defeat at local rivals Torino. Despite Pirlo's goal, Juve fell to a 2-1 defeat and missed out on wrapping up their fourth straight Serie A title . Still, Juve remain 14 points clear at the top of the league table and could still win a treble this season .
Baseline Abstractive: andrea pirlo scores a free free kick to sign a winner in the 2 - 1 defeat . the result means the <unk> are
Improved Abstractive: andrea pirlo scored a hat - kick for juventus , but could n't stop his side from a 2 - 1 defeat at local rivals
Questions: (1) Who scored the free kick? (2) Who does Andrea Pirlo play for? (3) Who were Juventus playing against?

Figure 1. Sample summaries produced by each of the models implemented for this study

After adapting the Bahdanau model to summarization on the CNN dataset, we received the results shown in Table 1 as our baseline results. Next, we implemented the beam search decoding method and varied the beam width parameter. Figure 2 shows the improvements made to ROUGE scores as a function of increasing beam width. Our final model's results shown in the table was our model with label smoothing and a beam width of 6.

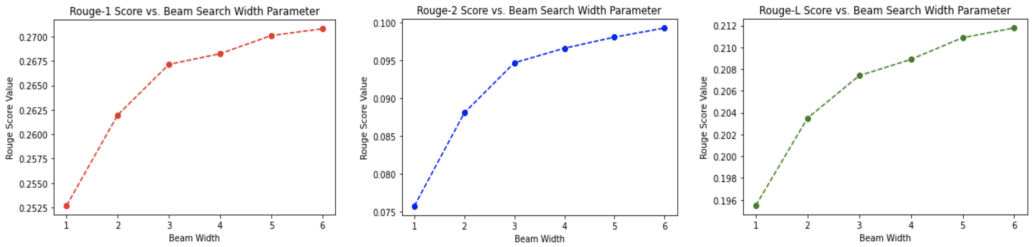


Figure 2. ROUGE score as a function of the beam width parameter for baseline model

We found that beam search significantly improved the results of our abstractive model in terms of ROUGE scores. However, from the plots in Figure 2 we can see that the improvement to ROUGE scores on each successive increase of beam width began to taper off after a beam width of 3. Due to the fact that increasing beam width causes summaries to take longer, this parameter should be optimized with respect to both computational efficiency and performance. Nevertheless, the utility of beam search decoding is clearly evident, and is strengthened by the fact that there are no signs of increasing beam width values past a certain extent leading to decreasing summary performance. Furthermore, we found label smoothing provided additional improvements to ROUGE score results. This is due to the fact that label smoothing tends to work in multi-class classification applications where there are many classes. Since our classes here is the length of the vocabulary, label smoothing is very beneficial during training of the network to prevent overfitting.

Model Type	Mean ROUGE-1	Mean ROUGE-2	Mean ROUGE-L	Mean QA accuracy
BERT	37.69	16.25	25.14	0.65
Baseline Abstractive	25.27	7.57	19.55	0.39
Improved Abstractive	27.66	10.56	21.49	0.43

Table 1. Comparison of ROUGE scores for each model configuration

Model Type	Number of questions right per summary	Mean confidence of correct answers	Mean confidence of incorrect answers
BERT	1.95	0.77	0.34
Baseline Abstractive	1.17	0.68	0.59
Improved Abstractive	1.28	0.64	0.58

Table 2. Comparison of Question-Answering metrics for each model configuration

Results from Table 1 show that we were not able to replicate state-of-the-art results in this study, as was expected given that our computational resources were a large limiting factor. Interestingly, however, our results demonstrate that ROUGE scores are not entirely correlated with the question-answering, specifically along the mean confidence question answering metric. As is evident from the results in Table 1 and Table 2 taken together, although the improved abstractive model was able to increase ROUGE scores significantly, this did not result in higher confidence answers to our questions. We hypothesize that this may be due to an increase in abstractiveness in the model following beam search decoding and label smoothing, but further work is required to test this. Ultimately, however, our results show that ROUGE scores and overall mean QA accuracy are correlated, as shown in Table 1, suggesting that our studies show no evidence of ROUGE scores failing to capture the quality of the summaries in this case.

In addition, we found that both the baseline abstractive model and the final model with label smoothing had lower average questions answered correctly in comparison to BERT. This indicates for both cases that the models were able to capture only one or two of the salient facts of the input text, on average. This was expected as the lack of computational power necessary to run these models contributed to their lack of accuracy. We also found that reduced dataset sizes lead to greater <unk> tokens during summary creation, especially for terms not commonly used in the entire corpus.

7 Discussion

Our results show the effect of recent improvements to state-of-the-art models, specifically through three techniques: label smoothing, adaptive learning rates, and beam search decoding strategies. The improvements to both ROUGE scores and mean QA accuracy between the baseline and improved abstractive models confirm the effectiveness of well-known improvements to natural language processing deep learning models according to not just one (ROUGE), but two key metrics (ROUGE and QA accuracy). Label smoothing is one improvement made to our baseline. We implemented label smoothing after reading papers by Lukasik et al. and Muller et al., which discussed label smoothing providing improved performance for deep learning tasks, including for seq2seq models such as ours [16][17] . Label smoothing modifies the softmax layer of the probability distribution for the predicted word by using soft, probabilistic targets rather than hard targets.

Beam search is another area we were particularly interested in exploring, as the improvements it makes over greedy decoding in natural language processing tasks are widely known and specific to decoder models like ours ([18], [19], [14]). As opposed to greedy decoding, which upon choosing the most likely word at any given time step and never looks back, beam search evaluates several different options (the number of options explored equal to the beam width) for the most likely word at each time step, and considers each option and all the options that follow before finally outputting the complete sentence with the highest likelihood. The decoder employed in our model is a conditional decoder, so each word output is conditioned based on the previous word and hidden state of the model. Thus, implementing beam search allowed our model to not be destined to output a bad sentence if it made a poor choice early on in the sentence. One caveat with beam search being that as beam width is increased, the computational time required to generate summaries also increases, which can elicit significant resource constraints.

Our secondary aim was to evaluate the use of question answering as a metric for summary performance. We found that our implementation of question answering somewhat correlated with ROUGE for the abstractive and extractive models, but it did not correlate with question answering confidence. This highlights the importance of future work in this field providing metrics other than the typically reported ROUGE scores as they may not capture the full story. It is important to note, however, that the confidence factor provided by the Hugging Face package did not accurately capture the accuracy of the model, since the confidence scores were not lower for questions answered incorrectly and higher for questions answered correctly. We experimented with an exponential decay learning rate scheduler on top of the Adam optimizer utilized by our improved and baseline abstractive models because we hypothesized doing so would allow for better convergence of the loss function to the global minimum. However we found that decaying the learning rate resulted in the learning rate not being aggressive enough and training perplexity plateauing.

Furthermore, we also experimented with LSTM's in the encoder and decoder, as opposed to bi-GRU's. However, LSTM's have more parameters than GRU's and are known to be slower / require more memory. Since we were already computationally constrained, we ultimately decided against pursuing this strategy. Additionally, we spent significant time trying to incorporate a distinct piece of the transformer from Vaswani et al.'s 2017 paper "Attention is all you need," namely a multihead attention layer as we hypothesized doing so would improve the quality of our abstract summaries [10]. We attempted to incorporate multihead attention via Rush et al.'s "Annotated Transformer" implementation of "Attention is all you need," but found that this would require fundamentally changing the structure of our existing baseline code and we were not able to successfully integrate multihead attention into our model in time. Similarly, we tried to evaluate the addition of Luong Attention, but also found this would require fundamentally changing the architecture of our baseline model, and we were not able to implement it [20].

In this paper, we successfully implemented an effective abstractive summarization model that was able to achieve comparable results to an extractive model after making various improvements to our baseline. Through this, we were able to learn what the necessary features of abstractive summarization model architectures are and how they have begun to be improved upon in recent years. This was exciting for our group as we had never before worked with summarization models. Future work should refine our abstractive text summarization model via modified attention layers and transformer architectures in order to improve the quality of outputted abstrative summaries for comparison to the state-of-the-art BERT model when evaluated via the question-answering task.

References

[1] Bahdanau et al., "Neural Machine Translation by Jointly Learning to Align and Translate," ICLR 2015

[2] Miller, D. (n.d.). Leveraging BERT for Extractive Text Summarization on Lectures. Retrieved December 15, 2020, from https://arxiv.org/pdf/1906.04165.pdf.

[3] Devlin, J., Chang, M., Lee, K., Toutanova, K. (n.d.). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Retrieved December 15, 2020, from https://arxiv.org/pdf/1810.04805.pdf.

[4] Lin, Chin-Yew. ROUGE: a Package for Automatic Evaluation of Summaries. In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, July 25 - 26, 2004.

[5] Rush, A., Nguyen, V., Klein, G. (2018, April 03). The Annotated Transformer. Retrieved December 10, 2020, from https://nlp.seas.harvard.edu/2018/04/03/attention.html

[6] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Advances in Neural Information Processing Systems, pages 1693– 1701.

[7] Sasikumar, U., Sindhu, L. (2014). A Survey of Natural Language Question Answering System. International Journal of Computer Applications, 108, 42-46.

[8] See, Abigail, Liu, Peter, and Manning, Christopher. 2017. Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368

Müller, R., Kornblith, S., Hinton, G.E. (2019). When Does Label Smoothing Help? NeurIPS. [9] Vasavada Vrinda and Bucquet Alexandre. 2019. Just News It: Abstractive Text Summarization with a Pointer-Generator Transformer.

[10] Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G.S., Hughes, M., Dean, J. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. ArXiv, abs/1609.08144.

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (n.d.). Attention Is All You Need. Attention Is All You Need. Retrieved December 9, 2020, from https://arxiv.org/pdf/1706.03762.pdf. Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. arXiv preprint arXiv:1606.02858.

[12] Eyal, M., Baumel, T., Elhadad, M. (2019). Question Answering as an Automatic Evaluation Metric for News Article Summarization. NAACL-HLT.

[13] Chen, D., Bolton, J., Manning, C.D. (2016). A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. ArXiv, abs/1606.02858.

[14] J. Bastings. 2018. The Annotated Encoder-Decoder with Attention. https://bastings.github.io/annotated$_e ncoder_d ecoder/$

[15] Brownlee, J. (2020, June 03). How to Implement a Beam Search Decoder for Natural Language Processing. Retrieved December 15, 2020, from https://machinelearningmastery.com/beam-search-decoder-natural-language-processing/

[16] Müller, R., Kornblith, S., Hinton, G.E. (2019). When Does Label Smoothing Help? NeurIPS.

[17] Michal Lukasik, Himanshu Jain, Aditya Krishna Menon, Seungyeon Kim, Srinadh Bhojanapalli, Felix Yu, Sanjiv Kumar. 2020. Semantic Label Smoothing for Sequence to Sequence Problems. EMNLP.

[18] Dive Into Deep Learning. (n.d.). Retrieved December 15, 2020, from https://d2l.ai/chapter$_r ecurrent - modern/beam - search.html$

[19] Wiseman et al., "Sequence-to-Sequence Learning as Beam-Search Optimization," cs.CL 2016

[20] Luong, M., Pham, H., Manning, C. D. (n.d.). Effective Approaches to Attention-based Neural Machine Translation. Retrieved December 13, 2020, from https://arxiv.org/pdf/1508.04025.pdf.