GUIR @ LongSumm 2020: Learning to Generate Long Summaries from Scientific Documents

Sajad Sotudeh[1], **Arman Cohan**[2] and **Nazli Goharian**[1]

[1]IR Lab, Georgetown University, Washington DC, USA
{sajad, nazli}@ir.cs.georgetown.edu

[2]Allen Institute for AI, Seattle, WA, USA
armanc@allenai.org

Abstract

This paper presents our methods for the LongSumm 2020: Shared Task on Generating Long Summaries for Scientific Documents, where the task is to generate *long summaries* given a set of scientific papers provided by the organizers. We explore 3 main approaches for this task: 1. An extractive approach using a BERT-based summarization model; 2. A two stage model that additionally includes an abstraction step using BART; and 3. A new multi-tasking approach on incorporating document structure into the summarizer. We found that our new multi-tasking approach outperforms the two other methods by large margins. Among 9 participants in the shared task, our best model ranks top according to ROUGE-1 score (53.11%) while staying competitive in terms of ROUGE-2.

1 Introduction

The task of document summarization aims at generating a short-form (summary) of a longer sequence of text (source) conveying the key points of the input text. This task can be generally performed in two ways: 1) Extractive: where the system finds the salient sentences within the source and concatenates them to form the summary (Zhou et al., 2018; Dong et al., 2018; Zhang et al., 2018; Narayan et al., 2018; Liu and Lapata, 2019; Xu et al., 2020); and 2) Abstractive: where the model conducts text generation, paraphrasing, and produces novel words that are not necessarily present in the source text (See et al., 2017; Çelikyilmaz et al., 2018; MacAvaney et al., 2019; Zhang et al., 2019; Raffel et al., 2019; Sotudeh et al., 2020a; Lewis et al., 2020).

Over the recent years, the task of summarizing scientific papers has attracted researchers' attention. This is due to linguistic challenges inherent to scientific domain and the longer length of the documents (i.e., scientific papers) in comparison with the documents in other domains such as news.

Most prior works in scientific summarization have focused on producing short-form summaries which are around 200 tokens per summary (Collins et al., 2017; Cohan et al., 2018; Xiao and Carenini, 2019) , rather than long-form summaries. Producing the summary at such length might be adequate when the source document is also of shorter form such as those in the news domain. Nevertheless, when summarizing longer documents such as scientific papers, producing short-length summary (i.e., abstract-like) more favors a high-level view of the source document, rather than covering all the salient information within a given source text. Producing such long summaries requires a deep and comprehensive understanding of specific scientific domain. Generating long summaries of the paper is helpful for researchers who might want to learn more about the paper beyond abstract-level information, without the need to read the entire paper.

The LongSumm 2020 shared task [1] aims to encourage the research at generating longer-form summaries for scientific papers, and we progress this challenge by our participation in this challenge, utilizing pre-trained transformer encoders for summarization task.

In our experiments, we explore three different methods for the challenge including 1) Experimenting with different versions of pre-trained transformer encoders finetuned for summarization task as an extractive method (Liu and Lapata, 2019); 2) A two-stage method where an abstarctive summarizer is added to the extractive summarizer to produce abstractive summaries; and 3) A novel multi-task learning model which aims at jointly incorporating documents' discourse structure into the extractive summarizer. This is inspired by the fact that having close attention to the scientific paper's

[1]https://ornlcda.github.io/SDProc/sharedtasks.html

Proceedings of the First Workshop on Scholarly Document Processing, pages 356–361

https://doi.org/10.18653/v1/P17

discourse information would result in improved summaries (Cohan et al., 2018). We report competitive results of our model, which achieves ROUGE scores of 53.11%, 16.17%, and 20.34% for RG-1, RG-2, and RG-L, respectively. With the obtained results, our system ranks 1st (RG-1), 2nd (RG-2), and 4th (RG-L) in terms of evaluation metrics.

2 Related Work

2.1 Scientific document summarization

Summarizing scientific papers has garnished vast attention from research communities during recent years, although it has been studied for decades. The characteristics of scientific papers, namely the length, writing style,and discourse structure, lends itself to some model considerations to tackle the challenging task of summarization. Researchers have utilized different approaches to address these challenges. For example, Cohan and Goharian (2015) utilized a citation-based approach, denoting how the paper is cited in the reference papers, to form the summary. Among the first large-scale datasets, Collins et al. (2017) introduced CSPubSum dataset, with the highlights of the paper (4-5 sentences) as the gold summaries. Cohan et al. (2018) introduced large-scale datasets of arXiv and PubMed, and used a hierarchical encoder to model the discourse structure of a paper, and then used an attentive decoder to generate the summary. More recently, Xiao and Carenini (2019) proposed a sequence-to-sequence model which incorporates both the global context of the entire document, and local context within the specified section. Yasunaga et al. (2019) introduced the first large-scale manually created scientific dataset, and proposed a hybrid method to integrate abstract and citations to form comprehensive summaries. Inspired by the fact that discourse information is of high importance when dealing with long documents (Conroy and Davis, 2017; Collins et al., 2017; Cohan et al., 2018) in scientific research papers, in this work, we step on utilizing such structure in summarization of scientific papers.

2.2 Pre-trained transformer networks

Due to the recent success of Transformer models such as BERT (Devlin et al., 2019), researchers have been motivated to fine-tune them on a variety of downstream NLP tasks such as text summarization. Liu and Lapata (2019) were the first to fine-tune BERT on summarization task. In their proposed model, they noted that since BERT outputs token-level vectors, it is not suitable for the extractive summarization task where the model often deals with sentences instead of tokens. To alleviate this problem, they appended a special `[CLS]` token to the start of each sentence to capture sentence-level representation, fulfilling the bases for extractive summarization task. Their model achieved the state-of-the-art on news domain. Later, BART (Lewis et al., 2020) was proposed which is an encoder-decoder pretrained Transformer model. For pretraining purposes, BART is trained by adding noise to the text, and then reconstruct the text by learning a model. In our model, we extend the BERTSUM model (Liu and Lapata, 2019) by adding a section predictor level that is jointly learned along with the sentence predictor layer (i.e., extractive summarizer).

3 Dataset

The dataset provided for this challenge consists of two types of summaries:

- Extractive summaries: these summaries are based on TalkSumm dataset (Lev et al., 2019), containing 1705 extractive summaries of scientific papers according to their video talks in associated conferences (i.e., ACL, NAACL, and etc.). Each summary within this corpus is formed by appending top 30 sentences of the paper. The average length of summaries in this corpus is around 990 words.

- Abstractive summaries: As an add-on dataset, the organizers have provided 531 abstractive summaries from different domains of CS such as Machine Learning, NLP, and AI, that are written by NLP and ML researchers on their blogs. The summaries' length in this dataset ranges from 100-1500 words per paper.

In our experiments, we use the extractive set along with 50% of abstracitve set as our training set, containing 1969 papers; and the other half of abstractive set is used as validation and test datasets. It has to be mentioned that the official test set (blind) also contains 22 abstractive papers.

4 Methodology

In this section, we discuss our methods with different configurations submitted to the shared task.

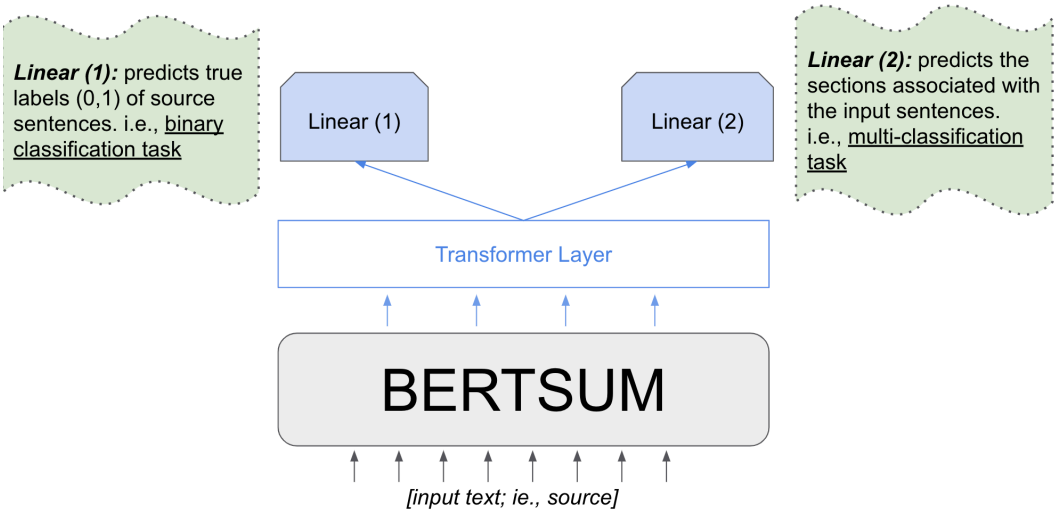

Figure 1: The overview of the proposed BERTSUMEXTMULTI model.

4.1 Pre-trained Transformers

While BERT has been shown to be effective on many natural language processing tasks, its application has not been straightforward for the text summarization task. This is due to the fact that BERT is trained with the objective of masked-language model, thus it results in token-level output vectors instead of sentence-level representations. This is particularly important since in the extractive summarization setting, the model needs to pick up sentences that are salient (Liu and Lapata, 2019). In our experiments, we utilize BERTSUM to obtain the BERT encoding for each sentence within the source document.

4.2 Extractive Summarization

As mentioned earlier, extractive summarization system aims at extracting top and salient sentences that are worthy to be included in summary. Let P show a scientific paper containing sentences $[s_1, s_2, s_3, ..., s_m]$, with m being the count of the sentences of input paper, and s_i denoting the i-th sentence in the document. The extractive summarization is then defined as the task of assigning a binary label ($\hat{y}_i \in \{0, 1\}$) to each sentence s_i in the input, deciding whether the sentence should be included in the summary or not.

BERTSUM is able to produce output vector t_i which is the representation of the i-th sentence within the input document. Afterwards, several inter-sentence Transformer layers are stacked upon top of BERTSUM outputs to collect document-level features for extractive summarization. The final output layer is a linear classifier with sigmoid activation function to decide whether the sentence should be included or not. In our experiments, we use this model to extract salient sentences (i.e., those with the positive label) to form the summary. We call this model as BERTSUMEXT.

4.3 Abstractive Summarizaiton

As the official test set provided for the challenge resembles more of abstractive type, rather than extractive, we aim at utilizing an abstractive model to produce abstractive summaries. After training the extractive model, we run the learned model through the entire papers in the dataset to extract salient sentences for each paper [2], resulting in a minimized input space for the abstractive summarizer. We then use BART, which is a denoising autoencoder for pretraining sequence-to-sequence model (Lewis et al., 2020), as the abstractive summarizer. In our experiments, we denote this model as BERTSUMBARTABS.

4.4 Section-aware Summarization

Inspired by few prior works that have studied the effect of document discourse structure in summarization task (Conroy and Davis, 2017; Cohan et al., 2018), we define a section prediction task, aiming at predicting section(s) that the sentences within the input documents belong to. Specifically, we add an additional linear output layer with sigmoid activation function that outputs scores for a set of pre-defined sections that a sentence can be assigned to. The entire extractive network is then trained to

[2] We used a threshold of 70 top sentences.

optimize both tasks (i.e, sentence prediction and section prediction) in a multi-task setting. For constructing training data for section prediction task, we take the approach and dataset introduced by Cohan et al. (2019) and run it over the sentences of the papers within the provided dataset to generate ground-truth labels. The overview of this model is shown in Fig. 1. This model is called BERTSUMEXTMULTI in our experiments.

4.5 Domain-tuned Summarization

Prior works have denoted the importance of fine-tuning language models on domain-related task and data (Gururangan et al., 2020; Sotudeh et al., 2020b). Following this paradigm, herein we experiment with fine-tuning summarization models on a sample of a larger dataset (i.e., arXiv (Cohan et al., 2018)) and an additional step of fine-tuning on the dataset provided in the LongSumm challenge. We use this scheme for both extractive and abstractive models (i.e., BERTSUM, and BART) with different settings. We call this model BERTSUMEXTMULTI-ARXIV.

5 Experimental Setup

As the initial parameters of the BERTSUM, and BART, we used the default hyper-parameters as denoted in the original papers (Lewis et al., 2020; Liu and Lapata, 2019). We used HuggingFace's Transformers library for working with BART [3], and also the open implementation for experimenting with BERTSUM [4]. In order to provide ground-truth labels for the task of section prediction, we utilized the external sequential-sentence package[5] by Cohan et al. (2019). It has to be mentioned that we classified the *Abstract* and *Conclusion* sections [6] into the same section without having their sentences labeled by the external package. For the joint model, we used loss weighting of 0.5 for two losses associated with each task as it resulted the highest scores in our experiments. For domain-tuned summarization, we used 50,000 samples of training set of arXiv dataset, and 2,000 papers samples from the validation set. In all our models, we pick the checkpoint that achieves the best RG-L score on the validation during training as our best model for inference. For submission purposes, we did a 5-fold cross validation on the second half of abstractive set, and report the average results of test sets over 5 different folds.

[3] https://github.com/huggingface/transformers

[4] https://github.com/nlpyang/PreSumm

[5] https://github.com/allenai/sequential_sentence_classification

[6] We used the title text matching to identify such sentences within Abstract and Conclusion.

6 Results

In this section, we present the performance of our submissions to the challenge, along with the scores achieved by the other participants. We then show the results of our systems over our internal test set that were constructed on the basis of abstractive set of summaries. Note that the reported scores are the average scores over the 5 folds of cross validation sets. To this end, we report the summarization systems' performance in terms of RG-1 (F1), RG-2 (F1), and RG-L (F1) metrics.

Table 1 shows the performance of our submitted systems to the challenge. For comparison, we also show results for the top 5 systems. As expected, our BERTSUMEXTMULTI model outperforms the other two models in terms of RG-1 and RG-L metrics. Comparing our best system's performance, we observe that our system outperform the other participants' system in RG-1 by large margin. While it lags behind the best submitted system by 0.9 point on RG-2 (i.e., comparable performance), and 1.04 point in RG-L.

Since the official test set is small, we also conducted analysis between variants of our model using the validation and an additional internal test set. We see in Table 2 that the Section predictor model performs fairly well over the model without section prediction module. This is particularly important finding since it characterizes the importance of document structure when summarizing a scientific dataset. Interestingly, having BART as the second stage does not yield to improvement in compared to the extractive setting. The most likely explanation of this gap is since the training set is biased toward extractive summaries, the BART model has difficulty figuring out how to produce right abstractive summaries (the number of abstractive summaries are limited in the training set), and in fact, the model learns to extract sentences, rather than producing novel words. We also trained BART on a portion of abstractive set as training set, but the performance was deteriorative, compared to the other settings. On the other hand, and interestingly, having summarization models fine-tuned on the external arXiv dataset does not

	RG-1	RG-2	RG-L
Other systems			
Summaformers	49.38	**16.86**	**21.38**
Wing	50.58	16.62	20.50
IIITBH-IITP	49.03	15.74	20.46
Auth-Team	50.11	15.37	19.59
CIST_BUPT	48.99	15.06	20.13
This work			
BERTSUMBARTABS	51.02	14.38	19.32
BERTSUMEXTMULTI-ARXIV	52.67	16.82	19.90
BERTSUMEXTMULTI	**53.11**	16.77	20.34

Table 1: ROUGE (F1) results of our submissions (bottom part of the table) to the challenge (official test set), along with the performance of other participants' systems. We only show top 5 participants in this table. Description of the other systems are not available at the time of submission. Please refer to the overview paper (Chandrasekaran et al., 2020) for details on each system.

	Validation			Test		
Model	RG-1(%)	RG-2(%)	RG-L(%)	RG-1(%)	RG-2(%)	RG-L(%)
BERTSUMEXT	45.39	12.41	17.81	45.34	12.42	17.82
BERTSUMEXTMULTI	**45.61**	**12.96**	**18.23**	45.55	**12.99**	**18.29**
BERTSUMEXTMULTI-ARXIV	45.44	12.95	17.99	**45.56**	12.77	18.06
BERTSUMBARTABS	44.88	11.78	17.63	44.44	11.51	17.26

Table 2: ROUGE (F1) results on abstractive set of LongSumm dataset (internal test set). The results are averaged over 5-fold cross validation.

yield much of improvement on this challenge. This might be due to the fact that the task defined on arXiv is for short summarization, not long which is our target task. For the section prediction task, BERTSUMEXTMULTI model achieves 92.3%, and 86.4% of accuracy in the validation and test sets, respectively.

7 Conclusion

In this paper, we approached the problem of generating long summaries given a scientific dataset of extractive and abstractive summaries. Our approaches explored using methods including 1) Pretrained transformer encoders for the extractive summarization task using BERTSUM; and 2) An abstractive summarizer (i.e., BART) which runs over the outputs of extractive summarizer at the first stage, to produce abstractive summaries; and 3) Our proposed novel multi-task learner where a section prediciton task is added to the extractive network, trying to jointly learn the sentence importance to be included in the summary, and the section associated with the sentence. While fine-tuning summarization model on external dataset does not yield promising results on this shared task, our best model is the one that jointly incorporates the section information into the extractive summarizer.

References

M. K. Chandrasekaran, G. Feigenblat, Hovy. E., A. Ravichander, M. Shmueli-Scheuer, and A. De Waard. 2020. Overview and insights from scientific document summarization shared tasks 2020: Cl-scisumm, laysumm and longsumm. *In Proceedings of the First Workshop on Scholarly Document Processing (SDP 2020)*.

Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Daniel S. Weld. 2019. Pretrained language models for sequential sentence classification. In *EMNLP/IJCNLP*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, W. Chang, and Nazli Goharian. 2018. A discourse-aware attention model

for abstractive summarization of long documents. In *NAACL-HLT*.

Arman Cohan and Nazli Goharian. 2015. Scientific article summarization using citation-context and article's discourse structure. In *EMNLP*.

Ed Collins, Isabelle Augenstein, and Sebastian Riedel. 2017. A supervised approach to extractive summarisation of scientific papers. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 195–205, Vancouver, Canada. Association for Computational Linguistics.

John M. Conroy and Sashka Davis. 2017. Section mixture models for scientific document summarization. *International Journal on Digital Libraries*, 19:305–322.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Yue Dong, Yikang Shen, E. Crawford, H. V. Hoof, and J. Cheung. 2018. Banditsum: Extractive summarization as a contextual bandit. In *EMNLP*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Guy Lev, Michal Shmueli-Scheuer, Jonathan Herzig, Achiya Jerbi, and David Konopnicki. 2019. Talksumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks. *ACL*.

M. Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, A. Mohamed, Omer Levy, V. Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ACL*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *EMNLP/IJCNLP*.

Sean MacAvaney, Sajad Sotudeh, Arman Cohan, Nazli Goharian, Ish Talati, and Ross W. Filice. 2019. Ontology-aware clinical abstractive summarization. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1013–1016, New York, NY, USA. Association for Computing Machinery.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *NAACL-HLT*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

A. See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.

Sajad Sotudeh, Nazli Goharian, and R. Filice. 2020a. Attend to medical ontologies: Content selection for clinical abstractive summarization. In *ACL*.

Sajad Sotudeh, Tong Xiang, Hao-Ren Yao, Sean MacAvaney, Eugene Yang, Nazli Goharian, and Ophir Frieder. 2020b. Guir at semeval-2020 task 12: Domain-tuned contextualized models for offensive language detection. *SemEval2020*, abs/2007.14477.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. *ArXiv*, abs/1909.08089.

Jiacheng Xu, Zhe Gan, Y. Cheng, and Jing jing Liu. 2020. Discourse-aware neural extractive text summarization. In *ACL*.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander Richard Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *AAAI*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Xingxing Zhang, Mirella Lapata, Furu Wei, and M. Zhou. 2018. Neural latent extractive document summarization. *ArXiv*, abs/1808.07187.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, M. Zhou, and T. Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *ACL*.

Asli Çelikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *NAACL-HLT*.