Convolutional Gated Unit for Improved Multi-Task Learning

CS224N: Natural Language Processing with Deep Learning
Default Final Project
Winter 2023

Daniel Contreras-Esquivel
Symbolic Systems Program
Stanford University
danielce@stanford.edu

Princess Vongchanh
Symbolic Systems Program
Stanford University
vongchan@stanford.edu

Abstract**—**Accessible summaries of scientific papers allow the average reader to better understand complex developments in scientific research. However, these summaries may lack sufficient information about the research and can even skew a reader's interpretation of the research if the original publisher's intended message is not conveyed well. BERT has proven to be effective for sentence-wise encoding, though it may struggle with longform texts. We hypothesize that fine-tuning the model for global encoding, specifically via a convolutional gated unit (CGU), will allow it to extract meaning from longform texts more effectively. To evaluate the impact that a CGU might have on a BERT model in general, we utilize a miniBERT model for multi-task classification. We found that implementing a CGU led the model to better recognize whether research summaries share the same sentiment and meaning as their corresponding research papers. We also found that overall, adding a CGU improves BERT performance. [1]

1. Introduction and Problem Description

Scientific research papers are vital for sharing new developments in research and are an objective source of information for individuals of all ages and levels of understanding who aim to learn about a topic. Because they are often complex–composed of crucial yet complicated concepts, specific terms, and even a certain tone or assumption of understanding in the writing–, literature reviews which summarize the research can minimize the barrier to learning that readers may face. However, these summaries may lack sufficient information about the research, its methods, and its findings; they may even skew a reader's interpretation of the research if the original publisher's intended message is not conveyed well. This project aims to gain an understanding of how effective literature reviews are at summarizing research papers and whether natural language processing models can generate summaries comparable to hand-written ones. In doing so, we develop a sequence-to-sequence (Seq2Seq) model for abstractive summarization.

Our research relies on pre-trained sentence embeddings from the Bidirectional Encoder Representations from Transformers, or BERT, language model. Although the Transformer's architecture is capable of handling long-term feature dependencies and BERT has proven effective for sentence-wise encoding, we anticipate that, when ideas are presented across multiple sentences in a single input, it may struggle to capture the nuances of how such sentences differ. It may also struggle with complex or unfamiliar vocabulary. The main goal of our research is to improve the model's ability to encode complex and longform texts. Specifically, we propose that the addition of a convolutional gated unit (CGU) will allow the model to better capture long-term features dependencies when evaluating and generating abstractive summaries.

2. Related Work

One specific understanding we entered this research with was that abstractive summarization models may misinterpret semantic meaning and therefore produce outputs with repeated phrases. Junyang Lin et al. (2018) [1] address this issue by proposing a global encoding framework which utilizes a convolution gated unit (CGU) to reduce repetition and improve the capturing of semantic meaning. The CGU is essentially a convolution neural network (CNN) that filters the outputs of a standard attention-based seq2seq encoder at each timestep, refining encodings with global context before being inputted to a decoder. A schematic of the authors' proposed CGU is included below (Figure 1). This is our research's main source of inspiration, and we adopt their proposal in both our multi-task classifier (see our Approach section for more details) and Seq2Seq model. Rather than develop our own encoder and decoder as they do, we fine-tune a BERT model downloaded from the Transformers library for the abstractive summarization portion of our research. Additionally, the dataset used to train our abstractive summarization model is much smaller than the GigaWord dataset they utilize with only 10 records. Junyang Lin et al. found that the addition of a CGU significantly reduced the degree of repetition in the abstractive summaries produced

[1] This template is based on the IEEE Computer Society template for conference papers.

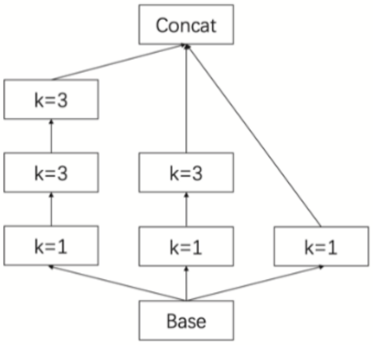

Figure 1. Structure of Junyang Lin et al.'s CGU which adopt [1]

by their model. We anticipate that our small dataset will result in underfitting; however, we expect that the model will still do better with a CGU.

We also found Siting Liang's and Klaus Kades's work particularly applicable, as they finetuned a BERT model for abstractive summarization of German radiology reports and we also sought to carry out our experiments on scientific literature. [2] The authors developed two different mechanisms which they tested on BERT2BERT models: The first mechanism followed an abstractive learning objective during the training inorder to get the model to generate a one summary sequence and extract the key points from the input source. The second mechanism used the implementation of a "pointer-network." This pointer mechanism allows the transformer-based seq2seq to decide between generating tokens from the vocabulary or from source sequence. The authors found that the model with both mechanisms outperformed a standard BERT2BERT model based on ROUGE score metrics. However, for the scope of this project, we will only be implementing a baseline version of the BERT2BERT model.

3. Approach

Overall, our project can be divided into three parts: building a multi-task classifier for evaluation, building the baseline abstractive summarization model, and implementing a CGU to both.[2]

3.1. Multi-task classifier

We first developed a multi-task classifier to measure the similarity of two texts. Three metrics of similarity are defined: how related their sentiments are (sentiment analysis), how related their semantic meanings are (semantic textual similarity analysis), and whether they are paraphrases of one another. Each of these tasks relied on pre-trained BERT embeddings, which we obtained via implementing a miniature version of BERT, or miniBERT. The tasks also required sentences to be encoded to obtain pooled representations of them. For the first metric, we applied dropout to the representations and projected the representations using a Linear layer; the output of this task were logits ranging from 1 - 5 that indicated the emotional tone of the sentences, 1 being negative and 5 being positive. For the second metric, we encoded pairs of sentences to extract the pooled representations of their semantic meanings and measure how close in meaning the sentences were using Cosine-Similarity; a float ranging from 0 to 1 was outputted for each sentence, where values closer to 1 indicated greater semantic similarity. For the third metric, we instead projected the pooled representations of a pair's semantic meaning using a Linear layer; this outputted one of two logits per pair, 1 indicating they are paraphrases of one another and 0 otherwise. The multi-task classifier was pre-trained with BERT embeddings and finetuned with single sentence inputs from the Stanford Sentiment Treebank and evaluated against the same dataset, as well as the SemEval STS and Quora datasets. Because our focus is on the effect of a CGU, we did not explore different ways to perform the initial multi-task classification tasks.

3.2. Baseline model: BERT2BERT model

We initially intended for miniBERT to be the encoder of our Seq2Seq model and faced several issues which are further discussed in Section blank. Instead, we relied on the Transformers library to build our model with support from a tutorial hosted on Google Colab by Microsoft's NLP-Recipes, a Github repository. We specifically imported from the Transformers library an EncoderDecoderModel, Seq2SeqTrainer, and a BertTokenizer. From the Hugging Face Datasets library, we used load_dataset to separately load the PubMed training dataset and testing dataset. The processing for these datasets was very minimal and all that was required was incorporating a function outlined in the tutorial which would help shape the datasets to match the model's inputs. We trained the baseline model and recorded the testing outputs for later evaluation. To finetune and test this model, we used the elife scientific_lay_summarisation dataset provided by Tomas Goldsack on Hugging Face.[3] The only pre-processing step for this dataset was to ensure that the data matches the inputs to our models including masks as needed.

3.3. Convolutional gated unit (CGU)

For our miniBERT model, we were interested in seeing whether this unit would improve sentiment analysis, paraphrase detection, or similarity prediction performance compared to baseline both when the CGU was added before or after finetuning. The unit was specifically added in the BertModel class' forward function. Our CGU took inspiration from the Jinyang et al., where we added a

[2] You can find our code here. The repository is public for this submission, and will be made private after March 27th.

[3] Thanks, Tomas Goldsack!

convolution neural network layer to the outputs from our encode function. We then performed ReLU after the convolution and passed this into our miniBERT's attention mechanism with the same extended attention mask used by the attention mechanism previously. We then multiplied the result of our CGU to previous encoding outputs resulting in the gating behavior and concluding the extent of our CGU.

Next, we used our miniBERT implementation as the encoder for a Seq2Seq model and from the Transformers library we imported a BERT decoder. During our research we found a few different repositories which implemented Seq2Seq summarization tasks and were able to use some of this code as the starting point for our summarization task [3], [4], [5], [6], [7], [8], [9]. However, in building our Seq2Seq model, we faced issues with integrating the code to the existing training and evaluation files. The Transformers library has a Seq2SeqTrainer class which we could have used to train/finetune the decoder, but unfortunately was incompatible with our miniBERT encoder.

For our BERT2BERT model, due to the libraries built-in trainer, we weren't able to implement the CGU as part of the model. However, we were still able to apply the CGU mechanism during the summary generation since we didn't use a library function for this. Here, we took the BERT encoder's output and passed it into our CGU as done previously for our miniBERT. We took the code from our miniBERT attention mechanism and integrated this to work for our CGU. In this case, we were not able to test whether the CGU being present during training would have an impact on performance, but we were still able to see whether the CGU would have an impact on performance during the summarization task post finetuning.

4. Experiments

As described in section 4, we developed two different models. First, we built a multi-task classifier and ensured it worked on the Stanford Sentiment Treebank, SemEval STS, and Quora datasets. Because our focus is on the effect of a CGU, we did not explore different ways to perform the initial multi-task classification tasks. After ensuring it worked, we tested it on the scientific_lay_summarisation (SLS) dataset, some of whoms inputs were truncated to accommodate for the maximum sequence lenghth that BERT accepts. Then, we added a CGU to the multi-task classifier and evaluated its performance using the same four datasets.

Then, we implemented a separate BERT2BERT model as outlined earlier. For this model, we were only able to add the CGU during the summarization generation task and not during finetuning. The abstractive summaries generated by this model were preprocessed similarly as the SLS dataset was in order to be tested by the multi-task classifier.

4.1. Evaluation Methods

We use the multi-task classification results as metrics to help assess whether our BERT-based seq2seq summarization model accurately captured the semantic meaning of the inputs. We had initially planned to incorporate performance on all three of the tasks into our final metric, but decided that the semantic textual similarity analysis encaptured our goal best. As you can see in Figure 2, we recieved low accuracy scores for semantic textual similarity with both iterations of the classifier. Regardless, we appreciate its greater interpretability as a float for each record and consider this a part of our final evaluation metric.

We also used ROUGE-1, ROUGE-2, and ROUGE-L scores given their widespread usage in evaluating the Seq2Seq model performance. The ROUGE library was downloaded to measure the number of similar n-grams. However, ROUGE scores are limited in their ability to capture semantic meaning because synonyms, which may or may not appear in the outputs, are not accounted for. To make up for this, we multiply the ROUGE scores with their respective degree of similarity for our final single-value metric. Thankfully, because the SLS dataset was only 10 records long, we were able to do this calculation manually for each record. We enter the project assuming the ROUGE score is 1 for exactly similar targets and references; therefore the final value represents how similar the corresponding documents are overall. Table 4 represents the average performance of the summaries according to the scoring metric below:

Where all ROUGE-1, ROUGE-2, and ROUGE-L $\in \{0, 1\}$,

$$S_1 = \text{ROUGE-1} * |\text{STS score}| \tag{1}$$

$$S_2 = \text{ROUGE-2} * |\text{STS score}| \tag{2}$$

$$S_L = \text{ROUGE-L} * |\text{STS score}| \tag{3}$$

We find ROGUE-1, ROUGE-2, and ROUGE-L F1 scores to be strong measurements of similarity because it considers both precision and recall between target and reference summaries. ROUGE-1 measures unigram similarity, ROUGE-2 measures bigram similarity, and ROUGE-L identifies the longest matching sequence of words.

5. Results

We found that applying the CGU to the multi-task classifier improved its performance on all four datasets (Figures 2 and 3), though disproportionately. We believe this is due

Dataset	Standard	With CGU
SST	0.513	0.612
Quora	0.5999	0.647
STS	0.218	0.298

Figure 2. SST, Quora, and STS dev accuracies

Task	Standard	With CGU
Sentiment	0.316	0.391
Paraphrase	0.190	0.201
Similarity	0.081	0.083

Figure 3. Scientific laysum summarisation dataset dev accuracies

to the difference in size between the latter dataset and the others. Also, the dev accuracy for semantic textual similarity of the SLS the input pairs used for paraphrase detection and semantic textual similarity analysis contained truncated research paper texts of 512 tokens each, while their corresponding literature review summaries had approximately 100 tokens each. The size difference for the pairs results in better performance on the paraphrase detection task, potentially due to its nature of outputting boolean values rather an a float score.

Our BERT2BERT model with the added CGU on the decoder outputs performed better than our control although the increase was not substantially significant.

We were surprised to see that the added CGU does not necessarily lead to an improvement in performance across tasks. However, this can be understood due to the mechanism we adopted was originally designed for machine-translation problems. Because of this, the CGU doesn't necessarily target the type of tasks we ran on our miniBERT. On the other hand, the CGU increased the performance of our BERT2BERT model on the ROUGE metric but not significantly, see figure 4. We believe that this is mostly due to the fact that our BERT2BERT had not been pretrained on scientific literature. For example, our Appendix has a sample summarization and the actual abstract from the research paper. Instead, had we imported bioBERT, for example, our model might have performed better when finetuned with the PubMed dataset in both with and without the [10]. In this case, it would have been interesting to have tested the CGU on an imported model which had embeddings that worked better in the field for which we were applying the summarization.

Base BERT2BERT	ROUGE Scores
ROUGE1	02.046
ROUGE2	00.980
ROUGE-F	01.255

BERT2BERT +CGU	ROUGE Scores
ROUGE1	06.237
ROUGE2	01.048
ROUGE-F	01.859

Figure 4. Average ROUGE Scores for control and +CGU model, multiplied by 100

Task	Score +CGU
Sentiment	0.406
Paraphrase	0.214
Similarity	0.092

Figure 5. SST, Quora, and STS dev accuracies

SLS	GAS +CGU
3.597	2.523
2.361	2.242
1.786	1.710

Figure 6. Final scores for SLS dataset and generated abstract summaries, average ROUGE * STS scores

Finally, we calculate the final scores of both our generated summaries (with the CGU applied) and the literature review summaries to evaluate which of them was more efficient (Figure 6). This required us preproccess the generated summaries for use with the multitask classifier (with the CGU applied) as well. [4] Performance on all three tasks only marginally better than the performance of the SLS dataset; we believe this is also because of the same issue around different input sizes.

Our research experiments that hand-written summaries outperform those generated by a language model. However, the difference between them is less than we expected! Perhaps this is because our scoring metric was less than ideal.

6. Conclusion and Next Steps

Altogether, we managed to implement a functioning miniBERT model for multitasking and a functioning BERT2BERT model for abstractive summarization. We then tested the impact a CGU would have on the performance of these BERT-based models with different setups for each model. We saw inconclusive results in the CGU's impact on the miniBERT model during multitasking. Our experiments demonstrated that adding a CGU results in some improvement in multitasking and summarization, but more trials and datasets are needed in further validating this and optimizing the CGU implementation. Further testing is required in understanding the role that this CGU might play for the different tasks our miniBERT had to complete. It is possible that modifying the CGU might potentially optimize its impact on the model so that the model improves accuracy across tasks. In contrast, our BERT-based seq2seq model with the CGU performed slightly better than its control. The papers of (Junyang Lin et al., 2018) [1] did not test their CGU on the BERT2BERT model, and our different CGU did not seem to show better results compared to their model.

[4] At this point, we wished we had coded up the calculation...

However, it would be interesting if we instead placed our CGU so that its inputs are the outputs from the embedding layer and then its outputs were then multiplied to the outputs of the encoding layer. In this case, the CGU would have picked up more of the nuances from the embeddings versus from the encodings. In theory, the encodings are already heavily processed, and applying the CGU to the embeddings would have reintroduced more of the original input's weight. We also could have split the research paper text in the SLS dataset into different sections so that they weren't truncated, and used a method for combining their scores.

Appendix

Below is the abstract of a research paper from PubMed and the abstract generated by our model on the same publication [11].

1. Original Abstract

"background and aim : there is lack of substantial indian data on venous thromboembolism (vte) . the aim of this study was to provide real - world information on patient characteristics , management strategies , clinical outcomes , and temporal trends in vte.subjects and methods : multi-centre retrospective registry involving 549 medical records of patients with confirmed diagnosis of vte (deep vein thrombosis [dvt] confirmed by doppler ultrasonography ; pulmonary embolism [pe] by computed tomography , pulmonary angiography and/or v / q scan) from 2006 to 2010 at three indian tertiary care hospitals . results : acute dvt without pe , acute dvt with pe , and pe alone were reported in 64 (352/549) , 23 (124/549) , and 13 (73/549) patients , respectively . mean age was 47 (16) years , and 70 were males . h / o dvt (34) , surgery including orthopedic surgery (28) , trauma (16) , and immobilization ¿ 3 days (14) were the most common risk factors for vte . hypertension (25) , diabetes (19) , and neurological disease (other than stroke) (8) were the most common co - morbidities . most (94) were treated with heparin alone (82) or fondaparinux (2) for initial anticoagulation; low molecular weight heparin alone (5) or warfarin / acenocoumarol (76) for long - term anticoagulation . anticoagulant treatment was stopped because of bleeding in 2 (9/515) patients . mortality was 7 among patients diagnosed with vte during hospital stay versus 1 in those hospitalized with diagnosed vte . the annual incidence of dvt (pe) increased from 2006 to 2010 . Conclusion : acute dvt alone was responsible for the substantial burden of vte in indian patients . bleeding was not the limiting factor for anticoagulant treatment in most patients ."

2. Summary Generated by BERT2BERT +CGU

"purpose : to investigate to what degree the presence of hypertension (htn) and poor glycemic control (gc) influences the likelihood of having microalbuminuria (mau) among cuban americans with type 2 diabetes (t2d). methods : a cross - sectional study conducted in cuban americans (n = 179) with t8 - 10. results : magnetic resonance imaging (mri) were measured from a randomly generated mailing list purchased from knowledge - base marketing, inc. participants were also assessed with anti - - catenin and anti - rad6 antibodies. strong expression"

Team Member Contributions

Daniel coded the multi-task classifier and Seq2Seq models, added CGUs to both of them, and used the Seq2Seq model to generate abstractive summaries. Princess supported with debugging both version of the multi-task classifier, preproccessed the SLS dataset for use, and ran all experiments beside abstractive summary generation using the team's AWS EC2 instance. Both team members completed the initial miniBERT for sentiment classification portion of the project via peer-programming, contributed equally to the final report, and ideated the initial topic and direction for the project. They sought each other out for continuous support on their respective parts.

References

[1] Junyang Lin, Xu Sun, Shuming Ma, and Qi Su. Global encoding for abstractive summarization. In *ACL 2018*, 2018.

[2] Siting Liang and Klaus Kades. Fine-tuning bert models for summarizing german radiology findings. 2022.

[3] Ala Alam Falaki. How to train a seq2seq summarization model using "bert" as both encoder and decoder!! (bert2bert). 2023.

[4] Berrin Yanıkoglu Figen Beken Fikri, Kemal Oflazer. Semantic similarity based evaluation for abstractive news summarization. 2022.

[5] Henry Dashwood. Fine-tuning bert for abstractive summarisation with the curation dataset. 2020.

[6] Anubhav. Step by step guide: Abstractive text summarization using roberta. 2020.

[7] Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280, dec 2020.

[8] Haoyu Zhang, Jianjun Xu, and Ji Wang. Pretraining-based natural language generation for text summarization, 2019.

[9] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China, November 2019. Association for Computational Linguistics.

[10] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[11] Kamerkar DR;John MJ;Desai SC;Dsilva LC;Joglekar SJ;. Arrive: A retrospective registry of indian patients with venous thromboembolism. *Indian journal of critical care medicine : peer-reviewed, official publication of Indian Society of Critical Care Medicine*, 2016.