META REVIEW GENERATION

Sri Ram Pavan Kumar Guttikonda
Auburn University
szg0148@auburn.edu

Abstract

Meta-Review generation is relatively new task which involved active research during the past few years. Meta-Review can be viewed as closely related to abstractive summarization but is not simple summarization, it involves other information like decision. It is dependent on many other sub tasks like decision prediction and reviewer sentiment which makes it complex and challenging. The main idea behind this project is based on the intuition that Meta-Review will contain mostly those sentences which are similar to sentences common among the three reviews and additionally it will contain decision. In this research I focused first on analyzing the how the current abstractive summarization models like BART, Pegasus and T5 work for the purpose of this task. The reviews generated were evaluated on a metrics like Bert-f1, sem-f1 and ROUGE. The results show that BART is dominant over the other models for this task. There are few case studies, where we first build the extractive summary out of input and build the abstractive summary out of it. The corresponding results are evaluated are discussed in detail in the paper.

1 Introduction

Peer-Review system is widely used by well known conferences to evaluate the research paper based on the artifacts presented by different reviewers. The decision with regard to a paper is taken based on the artifacts presented. The peer-review system functions as follows. On submitting a research paper to one of the established conferences like ICLR, ACM NeurIPS and so on. The Peer-Review system consists of the area chair/program chair who assigns the submitted research paper to three independent subject experts. The three subject experts evaluate the paper and present their review back to the area chair/program chair. The area chair/program chair makes a decision whether the paper has to be accepted or rejected based on the reviews and leaves comments/meta-review for the author.

With the surge in number of conferences, number of submissions to different conferences and lack of enough number of subject experts, the task became complicated due to strict time frame. The quality of the review got degraded because of the time complications. This lack of time also brought bias into the system and quality of review system compromised.Thus an AI agent is necessary to tackle this short coming, where the AI agent should be smart enough to generate meta-reviews and help in accurate decision making. The whole purpose of this research is to study about different summarization methods and come up with a solution to the problem stated above.

In this research I looked at different methods of doing the summarization task. I started with exploring traditional abstractive summarization models like BART, Pegasus and T5. Analyzed the models and evaluated the models across different metrics like Bert-f1, sem-f1 and ROUGE. The results are discussed in the results section. Later, I experimented on solving the content selection problem similar to (Gehrmann et al., 2018) and implemented my own extractive summarization algorithm.I have implemented two extractive summarization algorithms Com-sentence and Sentence-Cluster,

which I will be talking about in the later sections. Later I fed the extractive summary to the models to generate the abstractive summary. The results obtained are promising, need more future work which is discussed later as well. In the next section I will be explaining about the earlier works present in this task and explain a bit about them and later on move to explaining my methods and results obtained for the same.

2 Related Works

Meta-Review generation is still a new task and it has very few works around it. Earlier works include MetaGen(Bhatia et al., 2020) and Decision-Aware Meta-Review generation(Kumar et al., 2021). MetaGen(Bhatia et al., 2020) works on generating an initial draft by selecting the k top sentences using the Random Walk Restart algorithm. The initial draft is used to predict the decision, the decision and the initial draft together forms the final draft which is then used for meta-review generation. On the other hand Decision-Aware MRG(Kumar et al., 2021) enforces a multi-encoder architecture, where the reviews are given as input to multiple encoders and the outputs from the encoder are used to predict the decision and also given as input to a single decoder. The decision predicted is given to the last layer of the decoder and is also a part of the loss function.

These are the only research papers present based on Meta-Review generation to the best of my knowledge. These papers have their own limitations which needs to be addressed. MetaGen(Bhatia et al., 2020) uses the meta-review data available to update the weights for the sentence graph generated through Random Walk Restart algorithm, this makes it biased to particular conference. Decision-Aware(Kumar et al., 2021) is highly dependent on the decision prediction, If the predicted decision is wrong it will have a huge effect on meta-review making the system sensitive to predicted decision.

Apart from this we have our traditional transformer models which can help in the task of summarization. To talk about a few we have BART, Pegasus and T5. BART(Lewis et al., 2019) is a denoising autoencoder for pretraining sequence-to-sequence models. PEGASUS(Zhang et al., 2020) model is pretrained using Gap Sentences Generation(GSG) and Masked Language Model(MLM). T5(Raffel et al., 2020) was built particularly for getting the maximum out of transfer learning, where the model is trained on data rich tasks. All the three models BART, Pegasus and T5 were pretrained on certain downstream tasks which include abstractive summarization on datasets like XSUM and CNN/DailyMail.

The upcoming projects are focused mostly on Decision-Aware approach. In my project I moved forward with the intuition that most of the meta-reviews are based on the common sentences among the reviews and tried to solves the contention problem similar to those methods mentioned in (Gehrmann et al., 2018), (Duan et al., 2019) and (Nan et al., 2021). The main philosophy with these works is that the transformer is not smart enough to capture the most important sentences in a document and include them in the summary, so a mechanism need to implemented to overcome this problem. The same philosophy is used in my work which is described later. I believed that meta-review is not based on decision of the process and this can be viewed from my experiments, where the BART model was able to generate the decision of the paper in the meta review, even tough there is no special attention to generate the decision or the decision is not given as a part of input. From this observation I strongly believe that decision need not be given special care but there are other factors along with decision which need to be considered, about which I described in the future works.

3 Experiments

3.1 Dataset

There is currently no pre-defined standard dataset for the task. Dataset is not present due to data confidentiality and copyrights issue. PeerRead is the only dataset which consists of the crawler scripts to get the data from the conference site, but the crawler scripts are outdated because of the transformation of conference websites. The past works in this regard like (Bhatia et al., 2020) and (Kumar et al., 2021) used the PeerRead dataset or crawled the OpenReview

website to get their data points. (Bhatia et al., 2020) scraped the OpenReview website and also used the ICLR 2017 data from PeerRead dataset to conduct their experiments, they have around 3000 data points in total. Coming to the (Kumar et al., 2021), they scraped the ICLR conference from OpenReview website and got around 7500 data points for their experiments. Over the past few years conferences have been moving towards the open review system and hence data is available on Openreview website.

I have written the crawler script to scrape the OpenReview website and I was able to get around 1200 data points from the ICLR 2022 conference. The statistics of the data say that the average length of the concatenated review is around 1014 tokens. The average length of the meta review is around 114 tokens. The average no of sentences in the meta review stand between 6-7 sentences. This data is used to fine tune models pretrained on CNN/DailyMail dataset.

The models like BART, PEGASUS and T5 are pretrained on downstream tasks like summarization. The summarization task for these models used the XSUM(Narayan et al., 2018) , CNN/DailyMail(Chen et al., 2016) and Multi-news(Fabbri et al., 2019) datasets. If we have a look at the statistics of these datasets, the CNN/DailyMail datset seems to be more similar one to my dataset. The CNN/DailyMail has an average input tokens of 741 and the average summary tokens are 56 tokens. The CNN/DailyMail dataset has around 300k data samples. CNN/DailyMail dataset is though huge, but the input and summary tokens are similar to the dataset I have. For this purpose I have taken the models which are pretrained on the CNN/DailyMail datset and tried to fine tune it using the datset I have to generate a bigger summary.

3.2 Methods

Traditionally there are two ways through which we can generate a meta-review, 1) Extractive Summarization 2) Abstractive Summarization.

Extractive Summarization is the selection of most relevant top k sentences from the input. In extractive summarization we don't have a text generated we will have the sentences which are present in the input to form a summary by selecting the few important sentences from the input based on the algorithm. Text Rank, Lex Rank and LSA are some of the algorithms used to obtain extractive summary. In my work I have define two extractive summarization methods and they are as follows.

Com-Sentences: Given three peer reviews, here I will select the common sentences from the three peer reviews by using cosine similarity. If the cosine similarity between two sentence embeddings is above a threshold(0.75 for this project) then two sentence are considered to be similar and they are added to the list of similar sentences. Sentence embeddings are obtained using Bart based models. Using this we will get a list of common sentences. The sentences which are not a part of the common sentences list are used to form uncommon sentences list and the summary of this uncommon sentences is obtained using hugging face summarization pipeline. Now we have both common sentences and a summary of uncommon sentences. Both of them are concatenated to form the extractive summary. The intuition behind this approach is that summary will be a combination of sentences that appears in at least two peer review, plus some uncommon sentences. The sentences which appear in at least two peer reviews should be treated as important and it has high probability to be part of the summary. One more assumption in this regard is that a reviewer try to be concise with his review and doesn't repeat sentences.

Sentence-Cluster: In this approach, I divide the input sentences into clusters and then select one sentence from each cluster to form an extractive summary. In this method firstly I get all the sentence embeddings using the Bert based models. Next I will assign the sentence embeddings to the clusters using the K-Means clustering algorithm. The no of clusters for this task is taken as one third the no of sentence embeddings, assuming that this will ideally pick one sentence from each peer review and there will be less scope for error. After assigning the sentence embeddings to the cluster one sentence is picked from each cluster to form extractive summary. Initially I decided to take the no of clusters as average no

Algorithm 1 Com-Sentences

1: Get the sentence embedding of all the input sentences and form a sentence-embeddings list
2: Initialize a com-sent list
3: **for** x in sentence-embeddings **do**
4: **for** y in sentence-embeddings **do**
5: **if** x and y not in com-sent list and cosine similarity between x and y greater than 0.75 **then**
6: add x and y to com-sent list
7: **end if**
8: **end for**
9: **end for**
10: get list of uncommon sentences which are present in sentence-embedding and not in com-sent
11: generate summary of the uncommon sentences
12: Concatenate the com-sentence with the summary of uncommon sentences to get the extractive summary of the input.

of sentences in the meta-review, but from the dataset I have the average no of sentences stand around six sentences. If I consider no of clusters to be six I there is high chance that the cluster will include sentences which are not similar. Again I am using this summary to give it models like BART which is pretrained on dataset CNN, which has average input tokens as 741. It would be difficult for me to fine tune the model with a low data points. Considering these two Issues no of clusters is taken to be one third of the no of sentence embeddings, assuming that the scope for non similar sentences to be part of the same cluster to be low.

Algorithm 2 Sentence-Cluster

1: Get the sentence embedding of all the input sentences and form a sentence-embeddings list
2: Apply K-Means algorithm to divide the sentences into clusters
3: Take one sentence from each cluster
4: Form an extractive review concatenating all the sentences obtained from the cluster

These are the two methods I have used for this research and I found that the extractive summaries are good enough either same or better than the abstractive summaries obtained by models like Pegasus. The results were discussed in the results section. The Com-Sentence is found to be a better method of the two methods which is discussed later as well.

Abstractive Summarization on the other hand is complex compared to extractive summarization. Abstractive Summary is just not the input sentences alone it uses the input sentences to generate new text out of it which is at human level with the current state of the art Transformer models. It is a complex task since we need to generate new text out of it. In abstractive summarization we pass the input to a seq2seq model and get the summary generated. We can do abstractive summarization using models like BART, PEGASUS and so on. For this project I tried to analyse three models and they are as follows.

BART: BART(Lewis et al., 2019) uses a standard seq2seq architecture with a bidirectional encoder and a unidirectional decoder. BART is trained by corrupting documents and then optimizing a reconstruction lossâthe cross-entropy between the decoderâs output and the original document. BART supports a wider range of pretraining tasks like Token Masking, Token Deletion, Text-infilling, Sentence Permutations and so on. BART is later fine tuned on tasks like Token Classification, Sequence generation and so on. The Sequence generation task is done on datasets like CNN/DM and XSUM. BART was able to achieve a 5.41, 6.61 PPL values on the CNN/DM and XSUM tasks respectively. The ROUGE 1 score obtained by BART on CNN/DM and XSUM is 44.16 and 45.14 respectively. For the Purpose of my project I have used BART Large which has 24 layers, with max input tokens 1024, 511M parameters and pretrained on CNN/DM.

Pegasus: Pegasus(Zhang et al., 2020) propose a new self-supervised pre-training objective for abstractive summarization, gap-sentences generation, and study strategies for selecting those sentences. In PEGASUS, important sentences are removed/masked from an input document and are generated together as one out-

put sequence from the remaining sentences, similar to an extractive summary. The pre-trained tasks used in Pegasus are Gap Sentence Generation(GSG) and Masked Language Model(MLM). The Downstream tasks for Pegasus includes Classification, Summarization and so on. The Datasets used are CNN/DM, XSUM, Multi-news and so on. The results obtained for the summarization task include, ROUGE 1 score of 44.17 and 47.21 on the CNN/DM and XSUM datasets respectively. For the purpose of this project I had used Pegasus large pretrained on the CNN/DM.

T5: T5 (Text-To-Text Transfer Transformer)(Raffel et al., 2020) is a unified framework transformer model that is trained in an end-to-end manner with text as input and modified text as output. T5 model is trained on the unlabelled large text corpus called C4 (Colossal Clean Crawled Corpus) using deep learning. C4 is the web extract text of 800GB cleaned data. The pretraining task for this model include Masked Language Model(MLM), where the task is to work on a fill in the blank approach. The model also have Downstream tasks which include Classification, Summarization and so on. The fine tuning of the summarization is done using the CNN/DM. The results obtained show a ROUGE 1 score of 43.52 on the summarization task. In this project I tried to use both T5-base and T5-Large models. There is no standard model which is fine tuned on CNN/DM in hugging face. The available model T5-CNN is proposed by a research enthusiast and it gives similar results with T5-base, so I am using T5-base for this project.

In many of the recent works including (Gehrmann et al., 2018) there has an identification of the content selection problem faced by the transformers. To solve this problem the approach inclines to creating a extractive summary from the input and then generating an abstractive summary using models like BART. The work(Bhatia et al., 2020) also follows a similar approach for the purpose of this task where they generate a draft of the input using the Random Walk Restart algorithm and generate the meta-review using the UniLM. Using Random Walk Restart has its own disadvantages which I described earlier in the paper. So, in a similar fashion of work, I am going to use my own extractive summary algorithms which I described above. I am going to use the Com-Sentence and Sentence-Cluster algorithms to generate the extractive summary and feed this summary to models like BART, PEGASUS and T5 to generate meta-review from it. The results are evaluated and discussed in the further sections.

3.3 Experiment setup and Case Studies

The earlier works MetaGen(Bhatia et al., 2020) and Decision-Aware(Kumar et al., 2021) have their own limitations which were listed above. Both the research papers used seq2seq transformer for the abstractive summarization part. Transformers generally lag in content selection, as discussed in the (Nan et al., 2021) and (Duan et al., 2019) papers. Transformers may not be smart enough to select those sentences which should be present in the meta-review. In this research I have 4 case studies where I initially worked on the Transformer like BART, PEGASUS and T5. Later I worked on fine tuning these models with the dataset I have and obtained the results. The next test case include applying Com-Sentence and Sentence-Cluster algorithm to get the extractive summary. The Last case study includes giving the extractive summary to the Transformer models and look at the generated output. To start with each case study we have.

Case 1: In this case study I had worked on looking at how the results obtained by the models without any fine tuning looks like. In case of BART, I have used the BART large model which was pretrained on the CNN/DM and this model is available on hugging face, The BART model contains 511 model parameters. I have used the BART large model to give me the feasibility to input 1024 tokens. From the statistics of my dataset I have an average of 1014 tokens at input so I used BART large to compensate for the same. The next model I have considered is PEGASUS. I used the google PEGASUS model which was pretrained on CNN/DM and available on hugging face. The PEGASUS have

667 model parameters. The last model I have used is T5 model, I have used the hugging face T5-base model. The reason for using the T5-base model is because it is readily available on hugging face and there are few parameters to optimize, like this model has only 237 model parameters. The T5-large model is not giving better result so I have used T5-base. BART and PEGASUS are pretrained on CNN/DM, I used models pretrained on CNN/DM because my dataset is similar to the CNN/DM which I discussed earlier. For the case of T5, there is no standard T5 pretrained on CNN/DM so I have used T5-base. Initially all the three models are evaluated based on the results generated without any fine tuning in this test case.

Case 2: In the second case study we are using models which are fine tuned on the dataset I have. All the three models BART, Pegasus and T5 have the same characteristics as with case 1, but the parameters of the model are unfreezed and fine tuned. For the case of BART model we have 511 model parameters where 11 model parameter are left unfreezed and rest of the parameters are freezed during the fine tuning process. In Pegasus model there are 667 model parameters in total out of which 17 model parameters are left unfreezed. In the T5 model there are 237 model parameter out of which 7 model parameters are unfreezed for fine tuning. In all the three model the training dataset for fine tuning consists of around 600 data points and testing dataset consists of around 450 data points. The models are first fine tuned in this setting and the obtained model is used for getting the results. The results obtained are comparatively better compared to the case 1 and the same will be discussed later.

Case 3: In the third case we are obtaining the extractive summaries using the extractive summary methods discussed in the methods section. The extractive summary of the inputs are obtained using the Com-Sentence algorithm and the Sentence-Cluster algorithm. The obtained extractive summary are evaluated against the original meta-review and the results obtained are promising because the results are close to the results obtained by the model. Even tough the results are pleasing but considering a human approach for the problem we want meta review to be a generated text rather than the sentences from the input itself. So this case study has its drawback in that regard.

Case 4: In this final case we are going to generate the abstractive summary of the extractive summary obtained. In case 3 we talked that extractive summary for a meta-review is not humanly acceptable. We need a smart mechanism to generate the meta-review from the extractive summary we obtained in case 3. For this purpose I used the transformer models and fine tuned them with the extractive summary data I have. The set up for fine tuning the models is same as that of case 2, the no of freezed and unfreezed model parameters remain the same, the no of training and testing samples remain the same. The only thing changing from case 2 is the data which need to be fine tuned, earlier the input is the three concatenated peer reviews, now the input is going to be the extractive summary obtained by applying the extractive summary algorithms to the input of three concatenated peer reviews. After this we will have three fine tuned model fine tuned on the extractive summary data. The fine tuned models are used for generating the meta-review. The generated meta-reviews are evaluated and results are reported.

4 Results and Discussions

ROUGE has been used as the metric of evaluation for most of the summarization tasks including all the works I described in this paper. ROUGE is a good metric for extractive summarization task but is not a good metric to evaluate text generated by abstractive summarization, which is also described in the Decision-Aware(Kumar et al., 2021) paper. ROUGE doesn't take into account the semantics of the sentence and hence it would not be a suitable evaluation metric to evaluate the abstractive summaries generated. Taking this into consideration most of the recent papers on summarization tasks are considering Bert-f1 score as a suitable evaluation metric. Even tough Bert helps in understanding the semantics of the sentence and provide better numbers than ROUGE, Bert score lags for this particular task. If we

Table 1: Results Obtained
This table provides the BERT-f1, Sem-f1 and ROUGE scores of different Methods which was discussed in this paper

Method	Bert F1 Score	Sem F1	Rouge 1	Rouge 2	Rouge L	Rouge L sem
BART	0.8481	0.3698	0.2574	0.0450	0.1568	0.1566
BART Fine Tuned	0.8598	0.4371	0.3357	0.0839	0.2089	0.2090
BART Com Sentence	0.8569	0.4131	0.3212	0.0831	0.2055	0.2054
BART Sentence Cluster	0.8571	0.4114	0.3206	0.0832	0.2066	0.2068
Pegasus	0.8403	0.3825	0.2257	0.0379	0.1466	0.1465
Pegasus Fine Tuned	0.8478	0.3853	0.2817	0.0506	0.1686	0.1686
Pegasus Com Sentence	0.8469	0.3871	0.2674	0.0503	0.1690	0.1689
Pegasus Sentence Cluster	0.8438	0.3704	0.2433	0.0437	0.1610	0.1610
T5	0.8448	0.3238	0.1334	0.0263	0.1042	0.1042
T5 Com Sentence	0.8293	0.3216	0.1769	0.0306	0.1198	0.1199
T5 Sentence Cluster	0.8062	0.2531	0.1278	0.0178	0.0932	0.0933
Com Sentence	0.8315	0.3809	0.2817	0.0545	0.1415	0.1416
Sentence Cluster	0.8259	0.3489	0.2687	0.0466	0.1363	0.1362

have a look at Table 1 we can see that the average Bert-f1 score for every method is greater than 0.80, which is not expected as original meta-review and generated meta-review doesn't look very close. It is because of this inappropriate high score given by Bert, sem-f1(Bansal et al., 2022) evaluation metric is considered for this task. Sem-f1 gave the most helpful results and the number given by Sem-f1 looks appropriate and abide closely to human level evaluation. This could be treated as the first finding for this project.

The results obtained by different experiments and methods is displayed in Table 1. From Table 1 we can see that BART fine tuned model produced the best results on all the evaluation metrics present. From the results mentioned in the BART(Lewis et al., 2019),Pegasus(Zhang et al., 2020) and T5(Raffel et al., 2020) paper, on the CNN/DailyMail datset Pegasus Large model which was pretrained on HugeNews Dataset and Bart Large model showed the best ROUGE 1 score. From this it doesn't seem much surprising that the BART got the best results because the BART is pretrained on CNN/DM dataset which is close to the dataset I have. Pegasus fine tuned model gave poor results compared to BART probable because the dataset is low and also Pegasus is expected to generate summary similar to Extractive summary because of its pretraining tasks involved. More data is needed in this respect to clearly say that Pegasus is working poor in this task. A point to be noted is that the paper (Kumar et al., 2021) used both BART and PEGASUS and they showed that BART produced better results than PEGASUS. From the results in the (Kumar et al., 2021) and from the results I have it can be said that BART has an upper had in this task and more analysis is required to show how dominant BART is in this respect.

If we have a look at the generated meta-reviews from the boxes available.

BART produced qualitative results. There is one example where BART was able to produce the actual correct meta-review from input, and the original meta-review was not intact to the input reviews. From that example it can be said that the meta-review process has a bias and sentiments of the reviewer which needed to be taken into account along with the peer reviews. If we have a look at all the models the fine tuned model is able to provide a better result because the generated review length got increased. Initially all the models were pretrianed on the CNN whose summary has an average of 56 tokens, but the average meta-review in my dataset has 114 tokens so fine tuning was able to increase the length generated and improved the sem-f1 score.

When we talk about he 4 cases in the case study section above. For Case 1 Pegasus produced better results. When we compare the fine tuned models as stated in Case 2, the BART fine tuned model came out to be the best results, Pegasus fine tuned didn't work as expected partly because of the less data available for fine tuning and Pegasus have a lot of model parameters which need to be fine tuned. Comparing case 1 with case 2, case 2 gave out better results

because of the reasons mentioned above.

BART Generated:
bertscore(f1) : 0.8624 sem-f1 : 0.5269
generated:This paper studies how to use visual information in learning policies for self-supervised robotic robots. The paper proposes a novel architecture that uses visual information to learn to control a robot. The reviewers agreed that this paper is an interesting contribution to the field. The main concerns raised by the reviewers are that the paper is not clear on the significance of this work, and that the experimental results are not clear. The authors did a good job in responding to reviewers' concerns and addressed most of them in the rebuttal.
original:The paper addresses vision-based and proprioception-based policies for learning quadrupedal locomotion, using simulation and real-robot experiments with the A1 robot dog. The reviewers agree on the significance of the algorithmic, simulation, and real-world results. Given that there are also real-robot evaluations, and an interesting sim-to-real transfer, the paper appears to be an important acceptance to ICLR.

If we talk about Case 3, in Case 3 we obtained the extractive summaries using the Com-Sentence and Sentence-Cluster algorithm. One observation which is to be noted is that The Com-Sentence and Sentence-Cluster obtained results close enough to PEGASUS and T5 models which is promising. The Pegasus model is expected to work similar to an extractive summary algorithm based on the pretraining objective of the Pegasus model, and my extractive summary algorithms are producing results close enough to Pegasus model which is a promising sign. This observation could be a basis to say that if we were able to get the abstractive summary from the extractive summary obtained it could lead to better results.

PEGASUS Generated:
bertscore(f1) : 0.8434 sem-f1 : 0.4636
generated:The paper proposes a novel architecture that can train visual-locomotion policies end-to-end, and demonstrated good navigation/obstacle avoidance/uneven terrain walking results in the simulation. The main weakness of the paper: Not enough baselines to compare with. The authors may also shed light on under which scenarios we should choose RL-trained robots over Spot. The authors may also shed light on under which scenarios we should choose RL-trained robots over Spot.
original:The paper addresses vision-based and proprioception-based policies for learning quadrupedal locomotion, using simulation and real-robot experiments with the A1 robot dog. The reviewers agree on the significance of the algorithmic, simulation, and real-world results. Given that there are also real-robot evaluations, and an interesting sim-to-real transfer, the paper appears to be an important acceptance to ICLR.

In Case 4, we will be providing the extractive summaries to the models and get the meta-review generated. This step was expected to give better results than Case 2, but the results were similar to Case 2 because of the fact that data was not sufficient enough to fine tune the model to give out better results. For case 4 we have the extractive summary as the input and the extractive summary will have less no of input tokens compared to the actual review. The CNN dataset has input as 750 tokens and output summary has 56 tokens, the model pre-trained on such large dataset expects to take in a large data to give the best review out, since the extractive summary is not large enough better output was not obtained. The fine tuning is not helpful for the models as well because of the less amount of data, it won't be sufficient enough to fine tune parameters of large models.

Pegasus was expected to give better results as in the paper(Zhang et al., 2020) it is mentioned that it would give better results with as low as 1000 data points, which is not happening in this scenario. This problem of fine tuning is not affecting in Case 2 because in Case 2 the input is long and close enough to CNN dataset, hence the fine tuning in Case 4 is more affected than in Case 2. The results are however not bad and in fact close enough to Case 2 results. If we have a large dataset the fine tuning could be better and produce better results.

T5 Generated:
bertscore(f1) : 0.8508 sem-f1 : 0.4305
generated: a blind robot may have to make a few failed trials before it knows the height of a stair. the paper will be much stronger by including some more concrete comparisons with the current state-based learning approaches. the paper has extensive experiments and in-depth analysis in simulation.
original:The paper addresses vision-based and proprioception-based policies for learning quadrupedal locomotion, using simulation and real-robot experiments with the A1 robot dog. The reviewers agree on the significance of the algorithmic, simulation, and real-world results. Given that there are also real-robot evaluations, and an interesting sim-to-real transfer, the paper appears to be an important acceptance to ICLR.

T5 produced the poor results out of all the three models because T5 was pretrained on a sentence level and T5 is not fine tuned to greater level because of the low amount of dataset. As a result of this T5 generated shorter summaries and the summaries generated are more extractive in nature than abstractive.

I obtained better results than the earlier works (Bhatia et al., 2020) and (Kumar et al., 2021), but I cannot exclusively say my work is better than theirs, because the datset used is different and I haven't checked how their models work on the dataset I have, this is left for future work. The results I hold currently are promising.

5 Future Works

As discussed in the above section definitely more data is required to give a better conclusion and better output. Apart from that future works include working on the earlier proposed works(Bhatia et al., 2020) and (Kumar et al., 2021) to compare the performances. All the current active research work tends towards making the transformer decision aware to generate better results, but models were able to generate the decision even without any special attention and not providing decision in input. This suggests that there are other parameters which need to be considered like Bias of the reviewer and Sentiments of the reviewer into account. There should be some form of human evaluation for the results obtained to get better analysis.

6 Conclusions

In this research I tried to propose a system capable of generating the meta-review. I started with Analysing the efficiency of Transformer models for this task and later on moved to the analysis of different cases. I presented 4 Cases which was studied and the results regarding the same were also discussed. We can conclude from the research that the results obtained until now looks promising and it gave few insights on what problem need to be focused in future and worked on. From the results obtained till now Fine tuned BART was able to give out the best results. In future with the help of more data, the extractive summary algorithms proposed here could be altered and used to generate the best meta-review on presenting it to fine tuned models.

7 Code

The code can be found in the following github link.

https://github.com/sriram6399/Meta_Review_Project

References

Naman Bansal, Mousumi Akter, and Shubhra Kanti Karmaker Santu. 2022. Multi-narrative se-

mantic overlap task: Evaluation and benchmark. *arXiv preprint arXiv:2201.05294*.

Chaitanya Bhatia, Tribikram Pradhan, and Sukomal Pal. 2020. Metagen: An academic meta-review generation system. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1653–1656.

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*.

Xiangyu Duan, Hoongfei Yu, Mingming Yin, Min Zhang, Weihua Luo, and Yue Zhang. 2019. Contrastive attention mechanism for abstractive sentence summarization. *arXiv preprint arXiv:1910.13114*.

Alexander R Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749*.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*.

Asheesh Kumar, Tirthankar Ghosal, and Asif Ekbal. 2021. A deep neural architecture for decision-aware meta-review generation. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 222–225. IEEE.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen McKeown, and Bing Xiang. 2021. Entity-level factual consistency of abstractive text summarization. *arXiv preprint arXiv:2102.09130*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.